



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA EN ELECTRÓNICA Y TELECOMUNICACIONES

**TESIS PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
ELECTRÓNICA Y TELECOMUNICACIONES**

AUTOR: ZAMORA CRUZ, DIEGO GERARDO

**TEMA: EVALUACIÓN DE LA ESTRUCTURA DE CONTENT DELIVERY
NETWORKS COMO METODO PARA LA DISTRIBUCIÓN DE
CONTENIDO**

DIRECTOR: ING. AGUILAR, DARWIN

CODIRECTOR: ING. SAENZ, FABIÁN

SANGOLQUÍ, MARZO 2015

UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE
INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

CERTIFICADO
Ing. Darwin Aguilar

Ing. Fabián Saenz

CERTIFICAN

Que el trabajo titulado "Evaluación de la estructura de Content Delivery Networks como método para la Distribución de Contenido", realizado por el Sr. Diego Gerardo Zamora Cruz, ha sido guiado y revisado periódicamente y cumple en su totalidad con las normas establecidas por la Universidad de las Fuerzas Armadas – ESPE.

Debido a que se trata de un trabajo de investigación se recomiendan su publicación.

Sangolquí, Marzo del 2015,



Ing. Darwin Aguilar
DIRECTOR



Ing. Fabián Saenz
CODIRECTOR

DECLARACION DE RESPONSABILIDAD

UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE
INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD DIEGO GERARDO ZAMORA CRUZ

DECLARO QUE:

El proyecto de grado denominado "Evaluación de la estructura de Content Delivery Networks como método para la Distribución de Contenido", ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas que constan al pie de páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de mi autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

Sangolquí, Marzo de 2015



Diego Gerardo Zamora Cruz

AUTORIZACIÓN

UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE
INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

AUTORIZACIÓN

Yo, Diego Gerardo Zamora Cruz

Autorizo a la Universidad de las Fuerzas Armadas – ESPE la publicación en la bibliografía virtual de la Institución, el trabajo titulado "Evaluación de la estructura de Content Delivery Networks como método para la Distribución de Contenido", cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

Sangolquí, Marzo de 2015



Diego Gerardo Zamora Cruz

DEDICATORIA

Dedico este proyecto de tesis a Dios y a mis padres. A Dios porque ha estado conmigo a cada paso que doy, cuidándome y dándome fortaleza para continuar enseñándome que una caída no es derrota y que se debe luchar con dignidad con el fin de alcanzar los objetivos planteados, a mis padres, quienes a lo largo de mi vida han velado por mi bienestar y educación siendo mi apoyo en todo momento. Depositando su entera confianza en cada reto que se me presentaba sin dudar ni un solo momento en mi inteligencia y capacidad. Gracias a todo su apoyo hoy he logrado conseguir un objetivo más de vida.

AGRADECIMIENTO

En el presente proyecto de grado quiero hacer un humilde y sentido reconocimiento a todas aquellas personas que en algún momento de mi formación profesional y humana contribuyeron con su enseñanzas y paciencia, tanto en el diario vivir como en los salones de clases, convirtiéndose en un gran apoyo frente a las distintas adversidades que se presentaron durante este largo camino estudiantil.

Agradezco mis padres y familia que siempre buscaron la forma de apoyarme y guiarme por el camino del bien para que siga luchando por mis metas, sin su paciencia y confianza esta etapa de la vida no se podría culminar con éxito por eso mis agradecimientos eternos.

Finalmente y no menos importante un sentido agradecimiento a Dios por haberme permitido llegar hasta este punto y haberme dado salud, ya que sin ella no se podría seguir adelante día a día cumpliendo mis objetivos.

INDICE DE CONTENIDO

CERTIFICADO	ii
DECLARACION DE RESPONSABILIDAD.....	iii
AUTORIZACIÓN	iv
DEDICATORIA.....	v
AGRADECIMIENTO.....	vi
INDICE DE CONTENIDO.....	vii
INDICE DE FIGURAS	x
INDICE DE TABLAS	xi
GLOSARIO.....	xii
RESUMEN	xiii
ABSTRACT	xiv
CAPITULO I	1
INTRODUCCIÓN	1
1.1 PLANTEAMIENTO DEL PROBLEMA	1
1.2 ESTADO DEL ARTE	2
1.3 OBJETIVOS	4
1.3.1 Objetivo General	4
1.3.2 Objetivos Específicos.....	4
1.4 JUSTIFICACIÓN	4
1.5 ALCANCE	5
1.5.1 INTRODUCCIÓN	6
1.6 MARCO TEÓRICO.....	9
1.6.1 EVOLUCIÓN DE CDN	9
1.6.2 Características y ventajas de CDN	11
CAPITULO II	13
CLASIFICACION DE LA ARQUITECTURA CDN.....	13
2.1 COMPONENTE DE LA ARQUITECTURA CDN.....	13
2.2 COMPOSICIÓN DEL CDN	17
2.2.1 Organización CDN.....	18
2.2.2 Servidores.....	20

	viii
2.2.3 Relaciones	20
2.2.4 Interacción de Protocolos	24
2.2.5 Contenido/Tipo de Servicios	27
2.3 DISTRIBUCIÓN DE CONTENIDOS Y GESTIÓN	29
2.3.1 Selección y entrega de Contenido	30
2.3.2 Colocación de Sustitutos	32
2.3.4 Externalización del Contenido	37
2.3.5 Organización y gestión de la caché	38
CAPITULO III	41
TÉCNICAS DE REPLICACIÓN DE CONTENIDO DINÁMICO, ESCALABLE Y EFICIENTE	41
3.1 Web Caching	41
3.2 CDN extraíble basado en Un-Cooperative.....	43
3.3 CDN insertable basado en Cooperative.....	45
3.4 Objeto de sistemas de Localización	47
3.4.1 Servicios de directorio centralizado (CDS) y Servicios replicadas Directory (RDS)	47
3.4.2 Servicios de directorio distribuidos (DDS).....	48
3.5 Multidifusión para difundir actualizaciones.....	52
CAPITULO IV	53
ENTREGA Y GESTIÓN DE CONTENIDOS.....	53
4.1 Sistemas para la entrega Contenido Web	53
4.1.1 Capas lógica de un sistema Web	55
4.1.2 Una simplificada CDN Arquitectura	58
4.1.3 Aceleración de la generación de contenidos y Entrega	60
4.2 Capa de Replicación Front-End.....	62
4.3 Capa de Replicación de Aplicación	66
4.4 Capa de Replicación Back-End	69
4.5 Perfil de usuario capa de replicación	72
4.6 Simulación de la arquitectura CDN	75
4.7 Análisis de resultados de la simulación	86
4.8 Equipamiento CDN.....	91
4.8.1 Decoder ATEME DR8400	91
4.8.2 Encoder Cisco DCM 9900.....	93

	ix
CAPITULO V	95
CONCLUSIONES Y RECOMENDACIONES	95
5.1 Conclusiones.....	95
5.2 Recomendaciones.....	97
BIBLIOGRAFÍA	98

INDICE DE FIGURAS

Figura: 1.1. Diferencia con y sin CDN	8
Figura: 1.2. Evolución CDN	11
Figura: 2.3. Componentes de la Arquitectura CDN	14
Figura: 2.4. Operación Arquitectura CDN.....	18
Figura: 2.5. Relaciones CDN.....	21
Figura: 2.6. Relación según el enfoque de Red	22
Figura: 2.7. Matriz de Proxy	23
Figura: 2.8. Malla de proxy	23
Figura: 2.9. Esquema Interacción de Protocolos.....	24
Figura: 2.10. Esquema Distribución y Gestión de Contenido	30
Figura: 2.11. Esquema selección y entrega de contenido.....	30
Figura: 2.12. Esquema Estrategia de ubicación de los sustitutos	33
Figura: 3.13. Arquitectura CDN extraíble basado en Un-Cooperative.....	44
Figura: 3.14. Arquitectura CDN insertable basado en Cooperative.....	46
Figura: 3.15. Servicio de directorio centralizado (CDS) y Un servicio replicado Directorio (RDS).....	48
Figura: 3.16. Un directorio distribuido (Tapestry)	51
Figura: 4.17. Capas lógicas de un sistema de Web	54
Figura: 4.18. Arquitectura CDN simplificada.....	60
Figura: 4.19. Posible mapeo de sistemas Web capas lógicas en una infraestructura CDN.....	62
Figura: 4.20. La réplica de la capa de aplicaciones para el usuario en el servidor Edge	63
Figura: 4.21. La réplica de la capa de aplicación en el servidor Edge.....	67
Figura: 4.22. La réplica de la capa de fondo en el servidor Edge.....	70
Figura: 4.23. La réplica de la capa de perfil de usuario en el servidor Edge.....	73
Figura: 4.24. En el CDNsim muestra política de redirección CDN	76
Figura: 4.25. Opciones para la topología de la red en CDNsim	78
Figura: 4.26. Sitio web y el tráfico generado por el usuario.....	80
Figura: 4.27. Configuración del comportamiento de los servidores sustitutos	79
Figura: 4.28. Creación del Bottle y donde se almacenará.....	84
Figura: 4.29. Resultados desplegados vía web.....	86
Figura: 4.30. Esquema Ejemplo de conexión simulador CDN.....	87
Figura: 4.31. Resultados de Tiempo de respuesta al aumentar la densidad de usuarios	88
Figura: 4.32. Gráfica que demuestra la cantidad de servidores que se debe usar para un determinado grupo de usuarios.....	89
Figura: 4.33. DR8400	91
Figura: 4.34. Cisco DCM 9900	92

INDICE DE TABLAS

Tabla: 4.1. Ejemplo de Archivo Router (random50).....	79
Tabla: 4.2. Ejemplo de archivo Web (web_site16000).....	81
Tabla: 4.3. Ejemplo de archivos Tráfico (trace_file50000).....	82
Tabla: 4.4. Resultados vistos del lado del cliente	88
Tabla: 4.5. Resultado del uso de servidores réplicas dependiendo del número de usuarios	89

GLOSARIO

CDN. Content Delivery Network en inglés

Streaming. Hace referencia a los archivos multimedia tales como vídeo y audio.

DNS. Domain Name System. Servidor de dominios. Base de datos distribuida a través de Internet.

Caché. Copia que mantiene una computadora de las páginas web visitadas últimamente.

ISP. Internet Service Provider: Organización que provee la conexión de ordenadores a Internet, ya sea por líneas dedicadas o por líneas conmutadas.

DoS. Denial Of Service (DoS), denegación de servicio, incidente en el cual un usuario o una organización se ven privados de un recurso que normalmente podrían usar.

Webcasts. Es un diseño de transmisión a Internet donde transmite un medio en vivo similar a un programa de televisión o una emisora de radio.

Proxy. Mecanismo que permite compartir una única conexión a una red externa entre todos los puertos de una red local.

HTTP. En inglés Hypertext Transfer Protocol. Protocolo de Transferencia de Hipertexto.

URL. Uniform Resource Locator. Sistema de direccionamiento estándar de archivos y funciones en Internet, especialmente en la WWW.

DHTML. Dynamic HTML

Cluster. Se aplica a los conjuntos o conglomerados de computadoras contruidos mediante la utilización de hardware comunes y que se comportan como si fuesen una única computadora.

BGP. Border Gateway Protocol

Hash. Es método para generar claves o llaves que representen de manera casi unívoca a un documento o conjunto de datos.

Anycast. Es una forma de direccionamiento en la que la información es enrutada al mejor destino desde el punto de vista de la topología de la red.

Edge. Enhanced Data Rates for GSM Evolution (Tasas de Datos Mejoradas para la evolución de GSM)

TTL. Tiempo de Vida o Time To Live

RESUMEN

Con el pasar del tiempo el uso del internet se ha vuelto una prioridad indispensable para todas las personas de todo ámbito social, ya sea para revisar el correo empresarial o para un momento de distracción en redes social. Por tal razón las personas que demandan algún tipo de servicio Web lo que desea es obtener buenos tiempos de carga y tener una excelente navegación, lo cual para el proveedor que presta dichos servicios requiere tener una infraestructura de red que permita satisfacer dichos requerimientos. Los proveedores de servicios han visto una solución aceptable y que se está difundiendo ya en distintas partes del mundo es el uso de as CDNs, las cuales nos permiten tener réplicas del servidor central cerca de los clientes con el fin de mejorar el tiempo de respuesta, uso de ancho de banda y optimización de la red. Por tal razón en este proyecto sea investigación de la arquitectura de las CDNs con el objetivo de conocer más afondo esta nueva solución que se está difundiendo en las grandes empresa que se dedican a la distribución de contenido.

PALABRAS CLAVES:

CDN

DISTRIBUCIÓN DE CONTENIDO

RED

SERVICIOS

ABSTRACT

With the passing of time using the internet has become an essential priority for all people of all social, either to check corporate email or a moment of inattention on social networks. For this reason people who claim some sort of Web service you just want to get good load times and have excellent navigation, which for the provider that provides such services requires having a network infrastructure that would meet these requirements. Service providers have been solved acceptable and is already spreading in different parts of the world is using as CDNs, which allow us to have replicas of the central server close to customers in order to improve the response time, use of bandwidth and network optimization. For this reason in this research project is the architecture of CDNs in order to learn more afond this new solution that is spreading in large enterprise engaged in the distribution of content.

KEY WORDS:

CDN,

CONTENT DISTRIBUTION

NETWORK SERVICES

CAPITULO I

INTRODUCCIÓN

1.1 PLANTEAMIENTO DEL PROBLEMA

En los últimos tiempos el campo de las comunicaciones ha experimentado un crecimiento exponencial en el tráfico de datos debido a la popularidad del internet. Día a día las redes de comunicaciones están creciendo aceleradamente, esto nos lleva a analizar de una manera objetiva los diferentes procesos que existen, la demanda que hay en el mundo y la forma como se distribuyen las redes de comunicación.

Cada minuto del día se implementan diferentes métodos para el manejo de las redes de comunicación en diversas áreas de la ciencia, es por eso que existen infinitudes de protocolos de redes que nos ayudan a interactuar con otras aplicaciones para poder obtener información de los diferentes contenidos que existen en Internet.

El éxito comercial de Internet y de los servicios electrónicos, junto con el estallido de uso de complejos contenidos de los medios en línea se ha allanado el camino para el nacimiento y el interés creciente en las redes de distribución de

contenido (CDN). La creciente demanda del usuario para las garantías en la calidad del servicio ha dado lugar a la necesidad de estudiar y desarrollar nuevas arquitecturas de red y tecnologías para mejorar el rendimiento percibido del usuario al tiempo que limita los costos pagados por los proveedores. Se han propuesto muchas soluciones para aliviar los problemas de cuello de botella y los más prometedores se basan en el conocimiento del contenido que tiene que ser entregado. Las infraestructuras de redes tradicionales "contenido-ciegos" no son suficientes para garantizar la calidad de servicio a todos los usuarios. Nuevos protocolos y soluciones integradas debe estar en su lugar tanto en la red y en el lado servidor para distribuir, localizar y descargar contenidos a través de Internet.

Content Delivery Network (CDN) se conforma principalmente de un servidor central al cual estarán conectados otros servidores secundarios (son una imagen del servidor central) y a su vez estarán conectados a un Router (dispositivo donde se conectan equipos finales de usuario) y equipos finales (PC).

El funcionamiento es básicamente la interacción de la información en la red de manera transparente entre el usuario y el servidor (imagen) de una manera más rápida y eficaz, de tal manera que se pueda obtener el 50% de velocidad de carga para que su contenido se pueda desplegar de una manera más rápida y eficaz para el usuario.

1.2 ESTADO DEL ARTE

El desarrollo comercial que ha tenido el Internet y los e-servicio, ha permitido el nacimiento y el crecimiento de las redes de entrega de contenido (CDN). El tráfico de Internet a menudo tiene dificultades de rendimiento característicos de un entorno no dedicado. La demanda de los usuarios por tener un servicio de calidad ha provocado la necesidad de estudiar y desarrollar nuevas

arquitecturas y tecnologías de red para mejorar el rendimiento percibido del usuario al tiempo que limita los gastos pagados por los proveedores. Se han propuesto muchas soluciones para solventar los problemas de cuello de botella y los más prometedores se basan en el conocimiento de los contenidos que tiene que ser entregado. Las infraestructuras de red tradicionales no son suficientes para garantizar la calidad de servicio a todos los usuarios en una situación de tráfico dinámico y cada vez mayor. “Nuevos protocolos y soluciones integradas deben estar en su lugar tanto en la red y en el lado del servidor para distribuir, localizar y descargar contenidos a través de Internet” [1].

Las copias de contenido en un CDN están distribuidas geográficamente, para permitir la entrega veloz e íntegra en cualquier ubicación donde se encuentre el usuario final: a través de los servicios de CDN. Al utilizar CDN la entrega del contenido al usuario estático es de manera más eficientemente, ya que se nota un beneficio en velocidad con que el usuario final percibe la respuesta total de la página.

Los contenidos duplicados pueden ser objetos web, objetos descargables (archivos multimedia, software, documentos), aplicaciones, streaming y otros componentes de internet (DNS, rutas, solicitudes de base de datos).

Algunos de las empresa que hoy cuentan con este tipo tecnología CDN para webs en el mercado Comerciales: Akamai, Level 3, Amazon. Y algunas versiones gratuitas de CDN son: cloudflare, Coral Content Distribution Network, coBlitz, FreeCast, MediaBlog, PeerCast, PPLive, PPStream.

1.3 OBJETIVOS

1.3.1 Objetivo General

Evaluar la estructura de Content Network Delivery como método para la distribución de contenido

1.3.2 Objetivos Específicos

- Realizar investigación documental del estado del arte acerca de Content Delivery Network (CDN)
- Investigar los escenarios de internetworking de CDN, la arquitectura y metodología.
- Investigar los protocolos, servicios y Aplicaciones que usa CDN
- Analizar la potencialidad y servicios que presta CDN para redes de nueva generación.
- Realizar una simulación para comprobar la teoría de CDN

1.4 JUSTIFICACIÓN

Los usuarios de Internet comúnmente se sienten frustrados por un bajo rendimiento y pueden decidir abandonar un sitio web o desconectar una sesión multimedia cuando experimentan dificultades de desempeño. Aunque los modelos centralizados siguen vigentes en la actualidad Internet, estas arquitecturas son pobres en términos de adaptabilidad y escalabilidad. De ahí es que se comienzan a desarrollar otros protocolos, tecnología, nuevas infraestructuras de red para

estar a la par de la demanda de velocidad y rendimiento que exige el usuario, una de estas tecnologías es el CDN del cual se está hablando mucho pero realmente no se conoce como realmente funciona internamente y cuál es su estructura.

Las CDN emergen como la solución al actual problema que presenta una web centralizada: lograr bajo tiempo de respuesta y mínima pérdida de información, moviendo el contenido de la información más cerca de los usuarios. El objetivo es lograr un equilibrio entre los costos en que incurren los proveedores de contenido web y la calidad de servicio para los usuarios finales.

La razón de ser de este proyecto es adentrarse en la estructura y funcionalidad de CDN, y así entender como CDN hace para realizar una red supuesta de computadoras que contenga réplicas de datos, colocados en varios puntos de una red con el fin de mejorar el tiempo de acceso y de descarga de datos de clientes por la red.

1.5 ALCANCE

El presente proyecto se enfocará en un análisis de la arquitectura y el funcionamiento de la Red de Entrega de Contenidos en su parte medular ya que los atributos estructurales de un CDN revelan el hecho de que componentes de la infraestructura CDN están estrechamente relacionados entre sí. Por otra parte, la estructura de un CDN varía en función del contenido / servicios que presta a sus usuarios. Dentro de la estructura de un CDN, se utiliza un conjunto de sustitutos para construir la infraestructura de distribución de contenidos, algunas combinaciones de relaciones y se utilizan mecanismos para re direccionar solicitudes de los clientes a un sustituto y la interacción de los protocolos se utilizan para las comunicaciones entre los elementos de CDN.

El análisis constituirá en la investigación de varios aspectos del CDN como la Arquitectura CDN, la composición CDN, Distribución de Contenidos y Gestión, Solicitud de enrutamiento y muchos aspectos que nos darán una idea más clara y concreta del funcionamiento de esta Arquitectura que hoy por hoy es una solución importante para grandes empresa las cuales requieren que la velocidad de carga y actualización de contenidos más eficiente.

Además se investigar la existencia de algún software que nos permita simular el comportamiento CDN el cual, en teoría, ayudará a reducir en gran medida el tráfico de información y así obtener más velocidad de envío, mediante la generación de imágenes un servidor central para replicar el contenido casi inmediatamente a los servidores más cercanos al cliente.

El CDN es imprescindible para dar servicio a usuarios de todo el mundo, ya sea para ofrecer la descarga de archivos o para implementar sitios y aplicaciones web. En Ecuador pocas empresas ofrecen este tipo de servicios una de ella es Level 3, por esos mediante este proyecto es una gran oportunidad para dar a conocer esta solución y se pueda aplicar en más sectores empresariales de nuestros país.

1.5.1 INTRODUCCIÓN

En los últimos tiempos el desarrollo del Internet ha tenido un gran crecimiento que pocos nos esperábamos, convirtiéndose en una herramienta indispensable para cualquier actividad profesional, académica o de ocio. Lo cual ha provocado que la red se llega a saturar teniendo cuellos de botella en servicios populares en la Web, en una página corporativa o en los mismos sistemas de gestión universitaria.

La constante evolución del Internet involucra nuevos desafíos en la gestión y entrega de contenido a los usuarios, en el cual las personas encargadas de prestar servicios deben buscar una solución oportuna para que el usuario no experimente ningún problema al momento de acceder a algún tipo aplicación o sitio Web.

Debido a estos nuevos desafíos nace la red de entrega de contenido de información, (CDN por sus siglas en inglés) con el objetivo de crear servidores virtuales cerca de los usuarios para que el acceso sea mucho más rápido y evitando los cuellos de botella, así dejar a un lado el típico sistema centralizado que hoy en día está presente en muchas empresas que prestan algún sistema de servicio en la Web.

CDN nació para distribuir contenidos más solicitados desde los servidores web más populares del Internet, un CDN es capaz de entregar todo tipo de contenido dinámico, incluyendo varias formas de transmisión de los medios interactivos.

Las empresas encargadas de proveedor el servicio de CDN se dedicadas a acoger en sus servidores el contenido de los proveedores de contenidos de terceros, para la creación de reflejos de dichos contenidos en varios servidores repartidos alrededor del mundo, y para la transferir las solicitudes de los clientes de una forma transparente al reflejo más cercano y con una latencia mucho menor.

Una manera de entender el funcionamiento de CDN es mediante un ejemplo muy típico en nuestros días es cuando queremos disfrutar de la transmisiones de una final futbol y para ello en la mayoría de los casos recurrimos al internet.

Sin CDN: Cuando llego la hora del inicio del partido de fútbol muchos usuarios accedieron a una determinada página Web que trasmitiría el partido, lo que sucedió fue que la página se saturo, ya no admitió más usuarios y los demás usuarios que deseaban ver el partido tenían que esperar que otro se desconecte para poder conectarse ellos, además de que la trasmisión era de muy mala calidad debido a la densidad de usuarios.

Con CDN: Cuando llego la hora del inicio del partido el servidor de la página Web que trasmitiría el partido contaba con varios servidores virtuales ubicados en distintos lugares y eran exactamente una copia del servidor central, lo que género que todas las personas tengan acceso al ver el partido y experimente una trasmisión continua con una buena calidad.

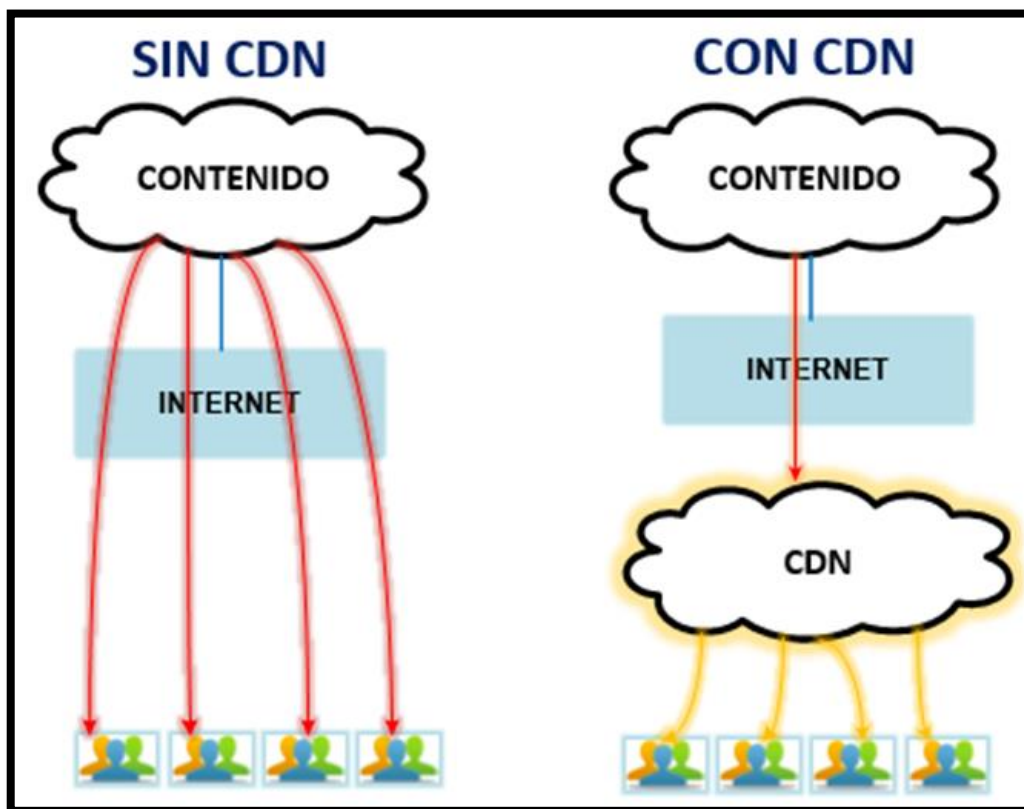


Figura: 1.1. Diferencia con y sin CDN

1.6 MARCO TEÓRICO

1.6.1 EVOLUCIÓN DE CDN

La red de entrega de contenidos surge de la necesidad de mejorar el rendimiento de la red mediante mejoras en la calidad de servicio:

- Una primera iniciativa se enfocó en mejorar la arquitectura de la Web, mediante el aumento la velocidad de procesamiento del servidor, más memoria RAM y espacio en el disco duro. Estas iniciativa no fue una mejora flexible ya que se llegó a un punto donde ya no se podía hacer más mejoras a la arquitectura y se veían en la necesidad de remplazar el servidor porque se tenía un crecimiento constante de usuario.
- Luego tenemos el almacenamiento del cache en el proxy por medio de un proveedor de servicios de Internet (ISP)¹, ya que para mejorar el rendimiento y reducir el uso de ancho de banda, el servidor proxy de almacenamiento en caché se encuentran desplegados cerca de los usuarios. Además están equipados con tecnología para detectar el fallo en el servido y maximizar el uso de los recursos de almacenamiento de cache de proxy. De esta forma direccionamos a los usuarios a un determinado contenido.
- Una solución que ha funcionado desde hace muchos años es la de ubicar servidores en distintos lugares geográficos de países, regiones, localidades, etc. Esta configuración es muy usada por muchos proveedores de contenido ya que se nota mejoras en el rendimiento y

¹ ISP significa literalmente Internet service provider (proveedor de servicios de Internet) o provider (proveedor). Éste es un servicio (en la mayoría de los casos pago) que permite conectarse a Internet.

el uso del ancho de banda, dicha configuración es conocida como granjas de servidores compuesta por varios servidores Web cada uno con una carga de usuarios determinada con un crecimiento más flexible y tolerante a fallas.

Pese a que la granja de servidores y el almacenamiento del cache resultan ser buenas soluciones para afrontar el uso de contenido Web en un cierto punto muestran algún tipo de limitaciones. Para hacer frente a estas limitaciones, surge otro tipo de red de contenido que se comenzó desplegarse a finales de 1990. “Esto se denomina como la Red de Distribución de Contenido, que es un sistema de ordenadores conectados en red a través de Internet a cooperar de forma transparente para la entrega de contenido a los usuarios finales”[2].

Con la evolución de CDN todos los proveedores se orientan a esta nueva solución ya que significativamente ver una reducción en costos de mantenimiento de infraestructura y una mejora significativa en la escalabilidad y una mayor fiabilidad, una de las empresa pioneras en esta nueva tecnología es Akamai Technologies la cual evolucionó a partir de un esfuerzo de investigación del MIT los cuales se enfocaron en el desarrollaron un conjunto de algoritmos de vanguardia para el enrutamiento inteligente y la reproducción de contenido a través de una gran red de distribución servidores.

Desde este momento CDN comienza una constante evolución a través del tiempo adaptándose a nuevos tipos de servicios como la voz y el video, a continuación se presenta el crecimiento que ha ido teniendo CDN en estos últimos años. (Figura: 1.2.)

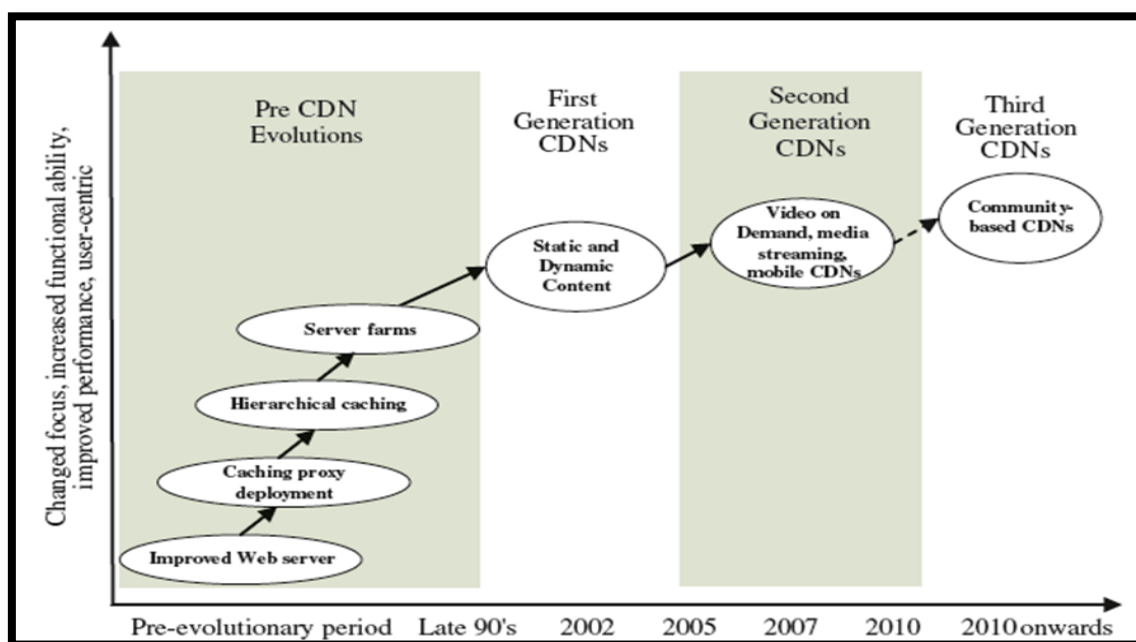


Figura: 1.2. Evolución CDN [2]

1.6.2 Características y ventajas de CDN

Las CDN son redes virtuales, cuyo direccionamiento y ruteo se basa en la descripción del contenido en lugar de su ubicación; tienen por lo tanto la flexibilidad de crear una topología virtual propia, no importando la red física subyacente: “por ejemplo, dos nodos conectados en la red virtual pueden estar muy alejados en la red física. Los nodos que pertenecen a dicha red virtual son llamados nodos de red o nodos de contenido” [3].

El objetivo que persigue el desarrollo de las redes de contenido es muy diversos, por las grandes ventajas que nos ofrecen para solventar necesidades particulares de nuevas aplicaciones con requerimientos específicos en cuanto a confiabilidad, disponibilidad, redundancia, performance, tiempo de acceso, anonimato, seguridad, etc.

- Las redes de contenido pueden adaptar su topología y en particular la cantidad y tipo de conectividad de sus nodos, logrando mayor o menor confiabilidad y mayor o menor latencia.
- La inexistencia de una dirección de alojamiento (IP) fija determinan menos vulnerabilidad a diversos ataques, en particular los del tipo de denegación de servicio (Denial of Service: DoS).²
- Se reduce la carga administrativa (operación y mantenimiento) de la red, cuando se requieren características de alta disponibilidad, redundancia y defensa contra ataques.
- Desde el punto de vista empresarial, las redes de contenido al utilizar aplicaciones propias del lado de cliente, permiten una fidelización por parte del mismo.

Cada uno de los servidores de un CDN contiene una copia de los archivos, y en el momento que se solicita uno, se sirve desde los servidores de la Red más próximo al usuario y/o menos congestionados, esto permite que, por ejemplo, un vídeo pueda ser visto por un número virtualmente ilimitado de usuarios simultáneamente. En el caso de las transmisiones en directo (Webcasts) el stream de vídeo se distribuye siguiendo el mismo criterio que los archivos bajo demanda.

² Es un incidente en el que un usuario u organización se ven privados de los servicios de un recurso que normalmente esperan tener.

CAPITULO II

CLASIFICACION DE LA ARQUITECTURA CDN

2.1 COMPONENTE DE LA ARQUITECTURA CDN

Los componentes de una arquitectura CDN básicamente son cuatro:

- Componente de entrega de contenidos: “Está conformado por un servidor de origen y un grupo de servidores replicas que entregan una copia de los clientes finales” [2].
- Componente de enrutamiento de solicitudes “es el encargado de dirigir las solicitudes de cliente a servidores de borde adecuados y para interactuar con el componente de distribución para mantener una vista de los contenidos almacenados en la caché de CDN” [2].
- Componente de distribución de contenido “mueve el contenido desde el servidor de origen para los servidores perimetrales de CDN y garantiza la coherencia de los contenidos en las memorias caché” [2].
- Componente de contabilidad mantiene registros de cliente accede y registra el uso de los servidores CDN. “Esta información se utiliza para

la presentación de informes de tráfico y la facturación basada en el uso por el propio proveedor de contenido o por una organización de facturación de terceros” [2].

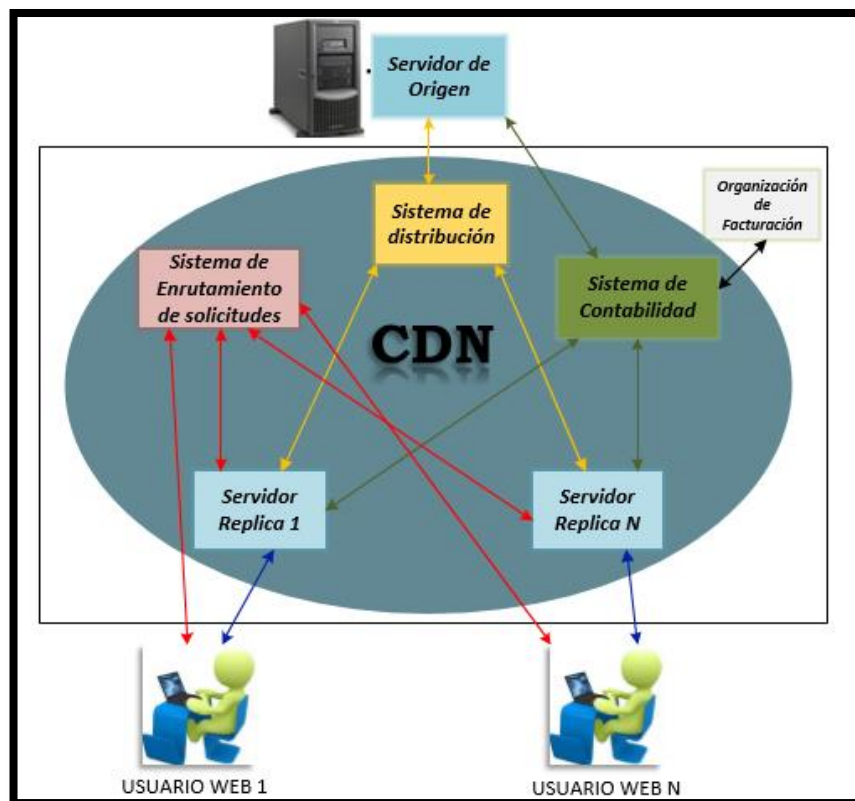


Figura: 2.3. Componentes de la Arquitectura CDN [2]

La interacción de estos componentes se realiza de la siguiente manera:

1. Los delegados de servidor de origen envían una “cadena de caracteres que identifica los recursos de una red de forma unívoca para los objetos del documento a ser distribuido y entregado por el CDN para el sistema de solicitud de enrutamiento” [4].
2. “El servidor de origen publica el contenido que se va a distribuir y entregado por el CDN en el sistema de distribución” [4].

3. El sistema de distribución de contenido se mueve a servidores de réplica. “Además, este sistema interactúa con el sistema de solicitud de enrutamiento a través de la retroalimentación para ayudar en el proceso de selección de servidor de réplica para las solicitudes del cliente” [4].
4. El cliente solicita los documentos de lo que percibe como el origen. “Sin embargo, debido a la cadena de caracteres que identifica los recursos de una red, la solicitud es en realidad dirigida al sistema de solicitud de enrutamiento” [4].
5. “La solicitud de enrutamiento de rutas del sistema da la solicitud a un servidor de la réplica adecuada en CDN” [4].
6. El servidor de réplica seleccionado proporciona el contenido solicitado al cliente. “Además, el servidor de réplica envía la información contable de los contenidos entregados al sistema de contabilidad” [4].
7. Los agregados del sistema de contabilidad separa la información contable en las estadísticas y los registros de detalles de contenido para su uso por el servidor de origen y la organización de facturación. “Las estadísticas también se utilizan como realimentación al sistema de petición de encaminamiento” [4].
8. “La organización de facturación utiliza los registros de detalles de contenido a un acuerdo con cada una de las partes involucradas en la distribución de contenidos y el proceso de entrega” [4].

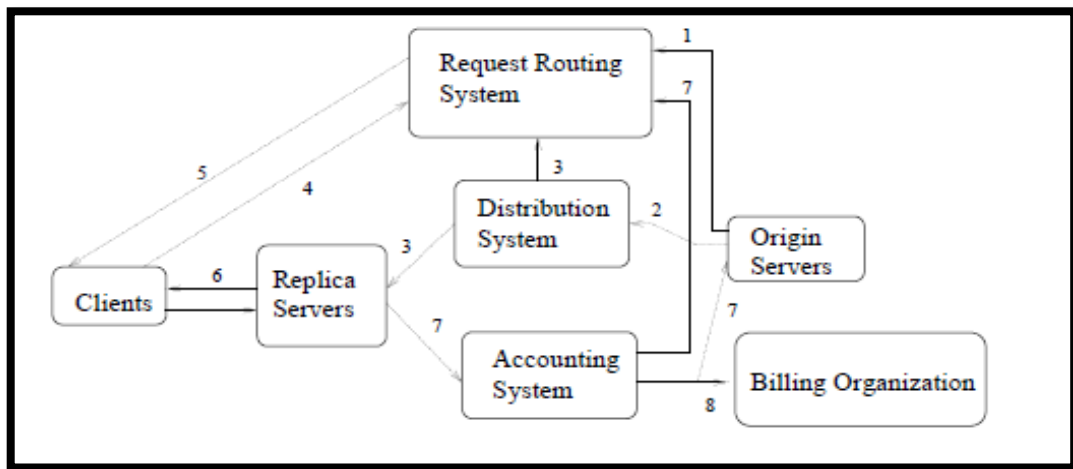


Figura: 2.4. Operación Arquitectura CDN [2]

La arquitectura de red de entrega de contenido se enfoca en la elaboración de una infraestructura que sea capaz de proporcionar los servicios de almacenamiento, gestión de contenidos, “distribución de contenido entre los servidores de borde, gestión de la caché, contenido dinámico de streaming³, soluciones de copia de seguridad y recuperación de desastres, medición del desempeño y la presentación de reportes del estado de los servicios” [2].

El cliente puede negociar con un proveedor de CDN para que su contenido Web sea colocado en los servidores de caché. Lo más común es que CDN acoja contenido de terceros incluyendo contenido estático es decir páginas HTML, imágenes, documentos, etc.

Las personas que requieren los servicios de CDN por lo general son las empresas de publicidad en Internet, medios de comunicación, centro de datos, tiendas de música online como iTunes, proveedores de servicios de internet (ISP). Cada cliente requiere que todo el contenido sea compartido con sus clientes finales de una manera eficiente y confiable.

³ Es la distribución digital de multimedia a través de una red de computadoras de manera que el usuario consume el producto, generalmente archivo de video o audio, en paralelo mientras se descarga.

La manera en que los proveedores de CDN facturan a sus clientes el servicio prestado es midiendo la carga de tráfico que presenta el contenido que se trasmite. CDN usa un mecanismo contable que da seguimiento a la información relacionada solicitada por el cliente final. Este mecanismo recolecta información en tiempo real y para cada componente CDN que se ve involucrado en la entrega del contenido al usuario final. Esta información es usada en la CDN para fines de contabilidad, facturación y mantenimiento. Los factores que determinan el costo de los servicios de CDN son:

- Uso del ancho de Banda
- La variación de la distribución del tráfico
- El tamaño del contenido replicado a través de servidores lo cual está relacionado con la carga de contenido que va tener cada servidor de borde.
- El número de servidores que captan la capacidad de un proveedor de CDN, es decir a cuantos lugares queremos llegar con nuestro contenido
- La fiabilidad y la estabilidad de todo el sistema de seguridad, directamente relacionado con el trato que se da a nuestro contenido sea el correcto y no se tenga inconvenientes con el usuario final.

2.2 COMPOSICIÓN DEL CDN

El estudio de las características de la estructura CDN nos indica que sus componentes están relacionados entre sí, y su estructura depende de la función del contenido/servicios que presta a sus usuarios. Dentro de la estructura de la

CDN utilizan un conjunto de sustitutos para construir el componente de contenido de la entrega.

A continuación mostraremos diversas características que componen la estructura de CDN, estos atributos son fundamentales para la composición de un CDN y se enfocan a la organización, tipos de servidores, las relaciones y las interacciones entre los elementos de CDN. (Figura: 2.4.)

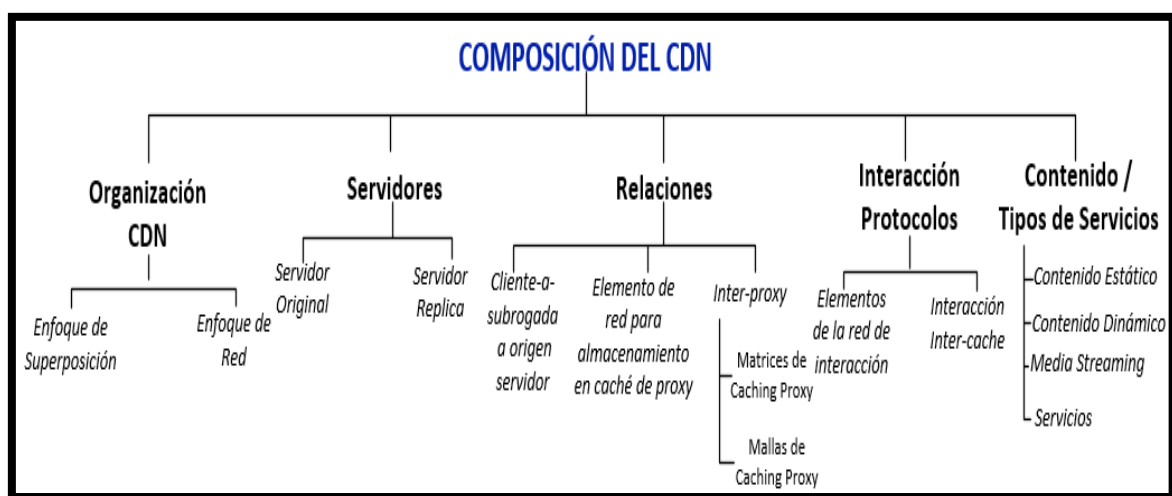


Figura: 2.4. Esquema de Composición CDN [2]

2.2.1 Organización CDN

La construcción de CDN se lo puede tomar desde dos puntos de vista: superposición y el enfoque de la red.

En el enfoque de superposición, los servidores de aplicaciones específicas y cachés están localizados en distintos lugares que maneja la red de distribución de contenido, el contenido que es transmitido puede ser Web, streaming media, y video en tiempo real.

Los equipos intermedios como switches y routers que se encuentran presentes en la red estrictamente se enfocan en la conectividad y en garantizar la Calidad de Servicio, en ningún momento intervienen en la entrega del contenido.

La mayoría de compañías dedicadas a prestar servicios de CDN usan el enfoque de superposición para la organización de CDN. Estos proveedores de CDN replican contenido a los servidores de caché en todo el mundo.

Cuando un usuario final solicita contenido este es redirigido al servidor CDN más próximo obteniendo así un mejor tiempo de respuesta en la carga de la Web. Si el proveedor de los servicios CDN ya no ve la necesidad de controlar los elementos de la red inferior, el enfoque de superposición se reduce permitiendo tener nuevos servicios.

En el enfoque de red, los elementos básicos que conforman la red entre ellos routers y switches están provistos con un código el cual le ayudan a la identificación de los tipos de aplicaciones y en la transmisión de las solicitudes basadas en políticas predestinadas.

En este tipo de enfoque los componentes de la red son los encargados de direccionar las solicitudes de contenido a caches locales o a servidores específicos los cuales se encargan de servir un tipo de contenido específico.

Algunos CDN usan los tipos de enfoques tanto el red y como el superposición de los para la organización de CDN. En tal caso, se puede aprovechar las ventajas de cada enfoque para mejorar la red de entrega de contenido colocando el elemento de red para que actúe en el extremo de una granja de servidores y redirija la solicitud de contenido a un servidor específico de la aplicación en el lugar más cercano al cliente final.

2.2.2 Servidores

La composición de una CDN cuenta con dos tipos de servidores: los servidores de origen y de réplica. El servidor en el que se encuentra la versión original del contenido se denomina servidor de origen y es actualizado por el proveedor del contenido. Por otro lado tenemos el servidor de réplica almacena una copia del contenido del servidor de origen, pero puede actuar como si fuera el principal para interactuar con los clientes. El servidor de origen se comunica con los servidores de réplica distribuidos en distintos lugares para actualizar el contenido que se encuentra almacenado en el mismo. El servidor réplica en CDN se lo usa como un servidor multimedia, servidor web o como servidor caché. Por el contrario el servidor de medios sirve cualquier contenido digital y codificado. Distintos tipos de servidores se encargan de atender las solicitudes de los clientes las cuales pueden ser un servidor de vídeo específico o clip de audio, un servidor Web contiene los enlaces a los medios de streaming o un servidor de caché los cuales tienen como objeto presentar el contenido el cual ellos disponen.

2.2.3 Relaciones

La arquitectura de un CDN presenta distintas relaciones entre los componentes que lo conforman. En estas relaciones se ven involucrados los clientes, sustitutos, almacenadores en cache del proxy, servidor de origen, entre otros elementos de la red. Estos componentes se encargan de la comunicación de la replicas y del contenido que se encuentra en la memoria cache dentro del CDN. La replicación del contenido con lleva a la creación y el mantenimiento de las copia duplicadas de un contenido dado en diferentes sistemas que almacenan y procesan la información.

En el medio de interacción de CDN, la relación básica para la entrega de contenido es uno de los clientes, sustitutos y servidores de origen. Un cliente

puede comunicarse con el servidor sustituto (s) para las solicitudes destinadas a uno o más servidores de origen. Cuando no se usa un sustituto, el cliente se comunica directamente con el servidor de origen. La comunicación entre un usuario y sustituto se lleva a cabo de manera transparente, como si la comunicación fuera con el servidor de origen previsto. “El sustituto sirve solicitudes de los clientes de su caché local o actúa como una puerta de acceso al servidor de origen” [2]. En la Figura: 2.5. podemos notar como sucede esta relación.

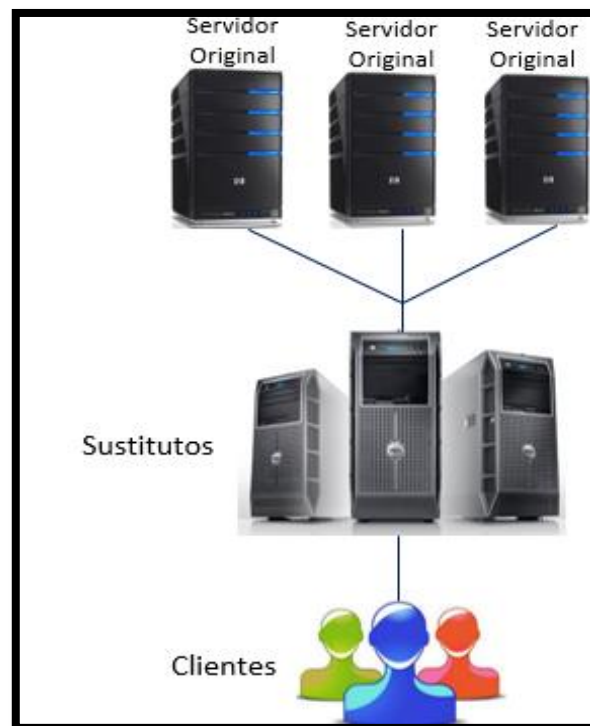


Figura: 2.5. Relaciones CDN

Como se mencionó anteriormente al usar el enfoque de red se usarían los elementos intermedios como router, switch para reenviar el tráfico a los servidores de almacenamiento que son capaces de atender las solicitudes enviadas por el cliente. “La relación en este caso es entre el cliente, elemento de red, y los

servidores de almacenamiento en caché / proxy⁴ o matrices” [2]. Como se lo representa en la Figura: 2.6.

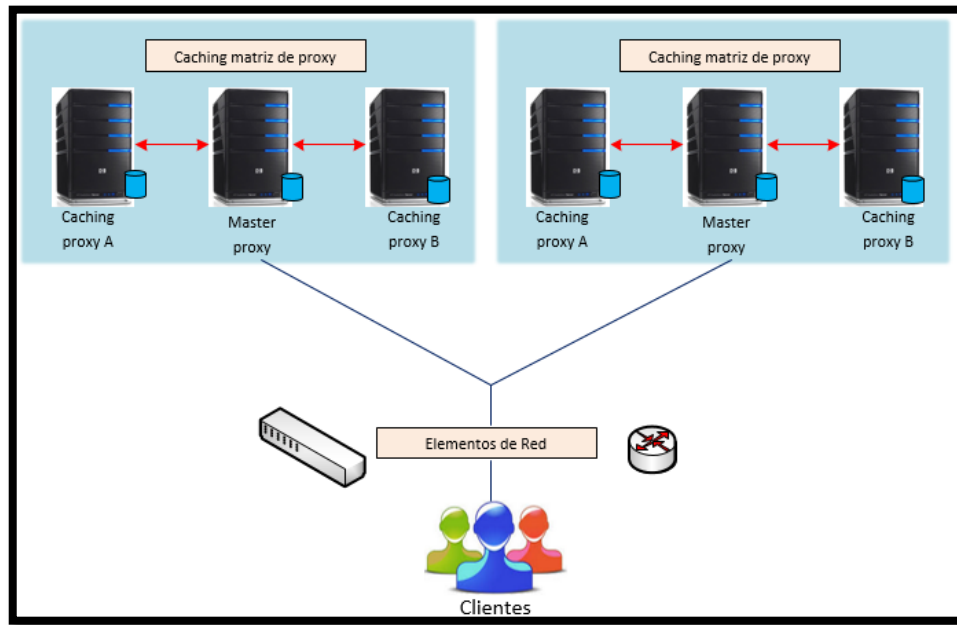


Figura: 2.6. Relación según el enfoque de Red [2]

Además de las relaciones mencionadas los proxies de almacenamiento en caché pueden comunicarse entre sí dentro de un CDN. Un proxy caché es un servicio de red de capa de aplicación para almacenar en caché los objetos Web. El Cachés de proxy se puede acceder y compartir por muchos usuarios al mismo tiempo. Una distinción clave entre las cachés de proxy CDN y cachés ISP es que el primero solo sirve contenido para un proveedor de determinados contenidos, es decir, los clientes de CDN, mientras que el contenido de la caché sirve a todos los últimos sitios web.

La comunicación inter-proxy los servidores proxy de almacenamiento en caché se pueden organizar de igual manera que unas matrices de proxy como se

⁴ Es un ordenador que sirve de intermediario entre un navegador web e Internet.

indica en la Figura: 2.7. y mallas de proxy como se muestra en la Figura: 2.8. En una matriz de proxy de almacenamiento en caché, un proxy autorizada actúa como un maestro para comunicarse con otros servidores proxy de almacenamiento en caché. Una malla proxy caché es una disposición de acoplamiento flexible de servidores proxy de almacenamiento en caché. A diferencia de las matrices de almacenamiento en caché de proxy, mallas proxy se crea cuando los proxies caché tienen relación uno-a-uno con los otros poderes. “Dentro de una malla de proxy de almacenamiento en caché, la comunicación puede ocurrir en el mismo nivel entre sus pares, y con uno o más master. Un servidor de caché actúa como una puerta de entrada a una malla como proxy y reenvía las solicitudes de clientes procedentes de proxy local del cliente” [2].

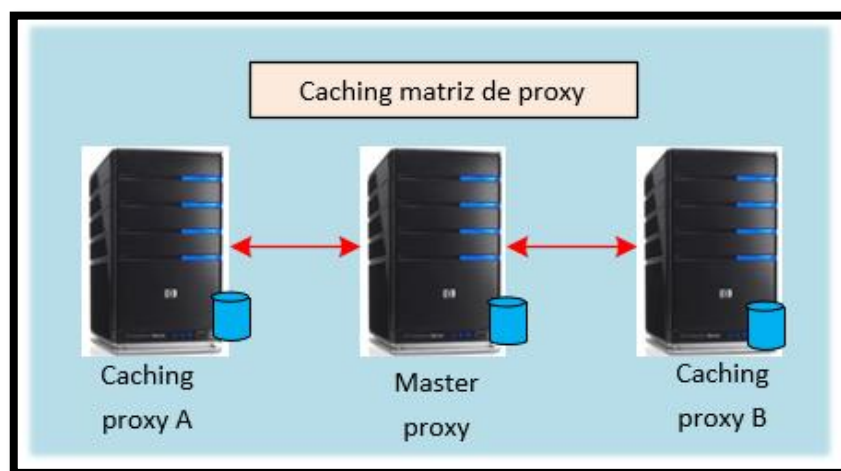


Figura: 2.7. Matriz de Proxy [2]

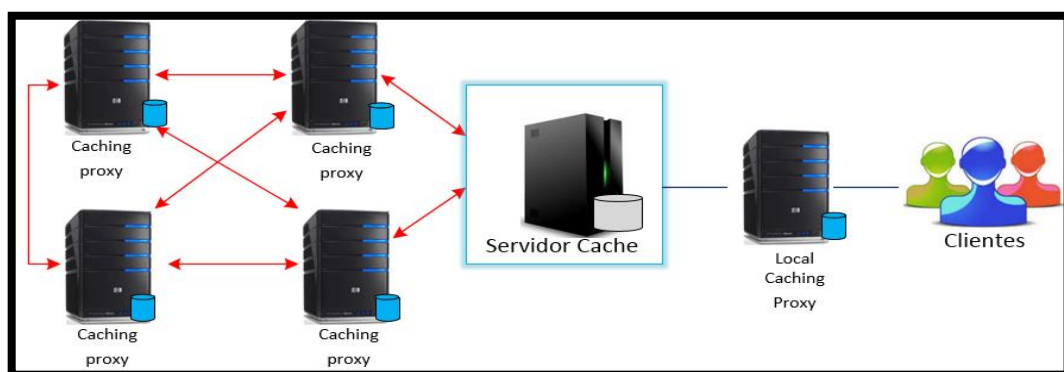


Figura: 2.8. Malla de proxy [2]

2.2.4 Interacción de Protocolos

Los protocolos que se usan para la interacción entre los componentes de CDN se los puede clasificar en dos tipos: La interacción entre los elementos de red y la interacción entre el cache, en la Figura: 2.9. se muestra los diferentes protocolos que usa CDN para la interacción entre sus elementos. “Algunos protocolos que se pueden mencionar están el Protocolo de Control (NECP) y Protocolo de control Web Cache. Por otro lado, la caché de Array Routing Protocol (CARP), Protocolo de caché de Internet (ICP), el protocolo de hipertexto Caching (HTCP), y Cache Digest son los ejemplos de protocolos de interacción entre la memoria caché” [2].

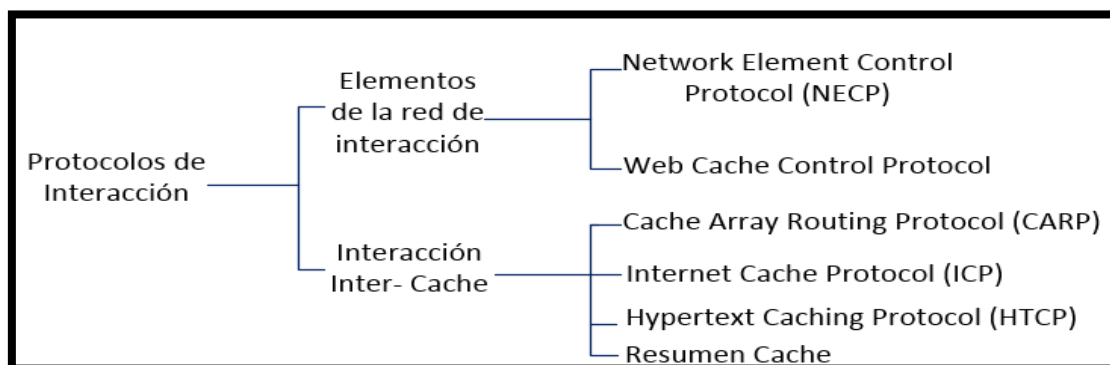


Figura: 2.9. Esquema Interacción de Protocolos [2]

- **El elemento Protocolo de Control de Red (NECP):** Es un protocolo ligero para la señalización entre los servidores y los elementos de red que reenviar el tráfico a ellos. Los elementos de la red consisten en una gama de dispositivos, incluyendo switches de contenido consciente y routers de equilibrio de carga. NECP permite que los elementos de la red para realizar balanceo de carga a través de una granja de servidores y la redirección a la interceptación proxies. Sin embargo, no dicta ninguna política específica de equilibrio de carga. “Más bien, este protocolo proporciona métodos para que los elementos de red puedan

aprender acerca de las capacidades del servidor, disponibilidad y sugerencias en cuanto a que los flujos pueden y no pueden ser atendidos. Por lo tanto, elementos de la red se reúnen la información necesaria para tomar decisiones de equilibrio de carga” [2].

- **Web caché Control Protocolo (WCCP):** Especifica la interacción entre uno o más routers y uno o más caches en la Web. Se realiza entre un router que funciona como un elemento de red de redireccionamiento y proxy de interceptación. El propósito de dicha interacción es establecer y mantener la redirección transparente para determinados tipos de flujo de tráfico a través de un grupo de routers. El tráfico seleccionado se redirige a un grupo de cachés en la Web con el fin de aumentar la utilización de los recursos y reducir al mínimo el tiempo de respuesta. Este tráfico incluye las peticiones del usuario para ver las páginas y gráficos de World Wide Web (WWW) en los servidores, ya sean internos o externos a la red, y las respuestas a esas peticiones. “Este protocolo permite que uno de los servidores proxy, sea proxy designado, para dictar al router cómo redirigir el tráfico a través de la matriz de proxy de almacenamiento en caché” [2].

- **Cache Array Routing Protocol (CARP):** Permite una gran variedad de equilibrar eficazmente la carga del cliente basado en la Web y distribuir el contenido en caché entre los miembros de la matriz. CARP ofrece equipos cliente con la información y los algoritmos necesarios para identificar el mejor servidor de la matriz para servir a su solicitud, lo que elimina la necesidad de que miembros de la matriz para reenviar las solicitudes entre ellos. “CARP también es compatible con la selección miembro de la matriz por los servidores y por los servidores proxy encadenados. CARP utiliza el enrutamiento basado en hash para

determinar el mejor camino para la resolución de una solicitud dentro de un arreglo” [5].

- **Protocolo de Caché de Internet (ICP):** ICP es un formato de mensaje utilizado para la comunicación entre cachés web. “Aunque las cachés web utilizan HTTP para la transferencia de objetos de datos, las cachés se benefician de un protocolo de comunicaciones más ligero y más simple. ICP se utiliza ante todo en una malla de cachés para localizar objetos web específicos en las cachés vecinas. Una caché envía una solicitud ICP a sus vecinos. Los vecinos contestan con respuestas ICP indicando un "acierto" o un "fallo"” [6].

- **El Hyper Text Protocolo Caching (HTCP):** Es un protocolo para descubrir cachés HTTP, los datos almacenados en caché, la gestión de conjuntos de cachés HTTP y el seguimiento de la actividad de caché. HTCP es compatible con HTTP 1.0. Esto está en contraste con ICP, que fue diseñado para HTTP 0.9. Mensajes HTCP pueden ser enviados a través de UDP o TCP. “Agentes HTCP no deben ser aislados de fallo de la red y los retrasos. Un agente HTCP debe estar preparado para actuar de manera útil en la ausencia de respuesta o en caso de respuestas pérdidas o dañados” [2].

- **Resumen Caché:** Es un formato de protocolo de intercambio y de datos. Proporciona una solución a los problemas de tiempo de respuesta y la congestión asociada con otros protocolos de comunicación inter-cache como ICP y HTCP. En cambio, otros servidores que asoman con ella buscar un resumen del contenido del servidor. Cuando se utiliza la Resumen Caché es posible determinar con precisión si un servidor concreto almacena en caché una URL determinada. Se realiza actualmente a través de HTTP. Aunque el uso

principal del Resumen caché es compartir resúmenes de las cuales URL se almacenan en caché por un servidor determinado, que puede extenderse a otras fuentes de datos. “Caché implícito puede ser un mecanismo muy poderoso para eliminar la redundancia y hacer un mejor uso de servidor de Internet y de los recursos de ancho de banda” [2].

2.2.5 Contenido/Tipo de Servicios

Las empresas dedicadas al negocio de CDN hospedan contenido de terceros para la entrega eficiente de contenido digital, streaming, contenido dinámico, contenido estático, entre otros servicios presentes en la Internet.

La variación en el contenido y los servicios entregados requiere un CDN para adoptar las características específicas de la aplicación, arquitecturas y tecnologías. Debido a esta razón, algunos de la CDN están dedicados para la entrega de contenidos y / o servicios en particular. Citar A continuación estudiaremos las características del contenido /tipo de servicios.

- **El Contenido Estático:** Se refiere a los contenidos para que la frecuencia de cambio sea baja. No cambia en función de las peticiones del usuario. Incluye páginas estáticas HTML, imágenes incrustadas, ejecutables, documentos PDF, parches de software, audio y / o archivos de vídeo. Todos los proveedores de CDN apoyan este tipo de entrega de contenido. “Este tipo de contenido se puede almacenar en caché con facilidad y su frescura se puede mantener el uso de las tecnologías de almacenamiento en caché tradicionales” [2].

- **El contenido dinámico:** Se refiere al contenido que es personalizado para el usuario o creado bajo demanda por la ejecución de un proceso de aplicación. “Con frecuencia cambia en función de las peticiones del usuario. Incluye animaciones, scripts y DHTML⁵. Debido a la naturaleza cambiante de la frecuencia el contenido dinámico, por lo general se considera como no almacenable en cache” [2].
- **Media Streaming:** Puede ser en vivo o bajo demanda. Entrega de medios en vivo se utiliza para eventos en vivo tales como deportes, conciertos, canal, y / o emisión de noticias. En este caso, el contenido se entrega "al instante" del codificador para el servidor de medios, y luego en el cliente de los medios de comunicación. En caso de entrega bajo demanda, el contenido se codifica y luego se almacena en forma de streaming de archivos multimedia en los servidores de medios. El contenido está disponible bajo las peticiones de los clientes de medios de comunicación. “En la demanda de contenido multimedia puede incluir audio y / o vídeo bajo demanda, los archivos de películas y clips de música. Servidores de streaming se adoptan con protocolos especializados para la entrega de contenido a través de la red IP” [2].

Los servicios prestados por CDN pueden ser directorios, almacenamiento Web, transferencia de archivos y servicios de comercio electrónico. Los servicios de directorio son proporcionados por el CDN para acceder a los servidores de bases de datos. “Consulta Usuarios para ciertos datos se dirige a los servidores de bases de datos y los resultados de las consultas frecuentes se almacenan en caché en los servidores de borde de la CDN” [2].

Servicio de almacenamiento web proporcionado por la CDN está destinado para el almacenamiento de contenido en los servidores de borde y se basa

⁵ (Dynamic HyperText Markup Language) es un término colectivo que sirve para designar una combinación de nuevas etiquetas del lenguaje HTML.

esencialmente en las mismas técnicas que se utilizan para la entrega de contenido estático. Servicios de transferencia de archivos facilitan la distribución a nivel mundial de software, las definiciones de virus, películas bajo demanda, y las imágenes médicas altamente detalladas. Todos estos contenidos son estáticos por naturaleza. Tecnologías de servicios Web son adoptadas por un CDN para su mantenimiento y la entrega. El comercio electrónico es muy popular para las transacciones comerciales a través de la Web. Los carros de compras para servicios de comercio electrónico pueden ser almacenados y mantenidos en los servidores de borde de la CDN y las transacciones en línea se pueden realizar en el borde de la CDN. “Para facilitar este servicio, servidores perimetrales CDN deben estar habilitadas con el almacenamiento en caché de contenido dinámico para sitios de comercio electrónico” [2].

2.3 DISTRIBUCIÓN DE CONTENIDOS Y GESTIÓN

La distribución y gestión de contenido en un CDN forma la parte primordial la cual permite que se tenga una eficiencia en la entrega de contenido.

La distribución del contenido es un ente importante dentro de la CDN ya que su labor es la selección del contenido, la frecuencia de solicitudes que envían los usuarios y la entrega del contenido dependiendo del tipo.

En cambio la gestión de contenido es usada para la organización, actualización y mantenimiento del cache. En la Figura: 2.10. mostramos un esquema general con cada característica de importantes para la Distribución y Gestión de contenido.

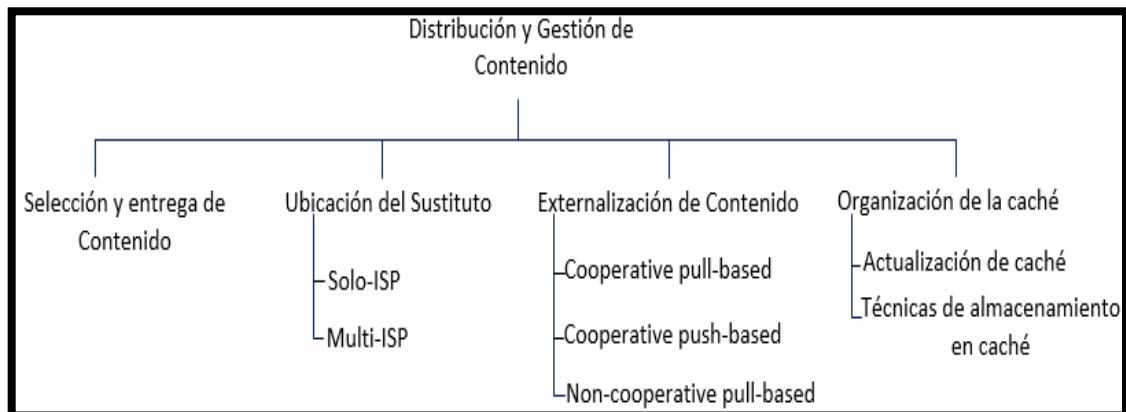


Figura: 2.10. Esquema Distribución y Gestión de Contenido [2]

2.3.1 Selección y entrega de Contenido

La clave del éxito de la entrega de contenido radica en la eficiencia en la selección del contenido que será entregada al usuario final. Una perspectiva adecuada en la selección del contenido nos ayudará a mejorar en el tiempo de descarga por parte del cliente y la carga del servidor. En la Figura: 2.11. a continuación mostramos las diversas técnicas que son usadas para la selección de contenido.

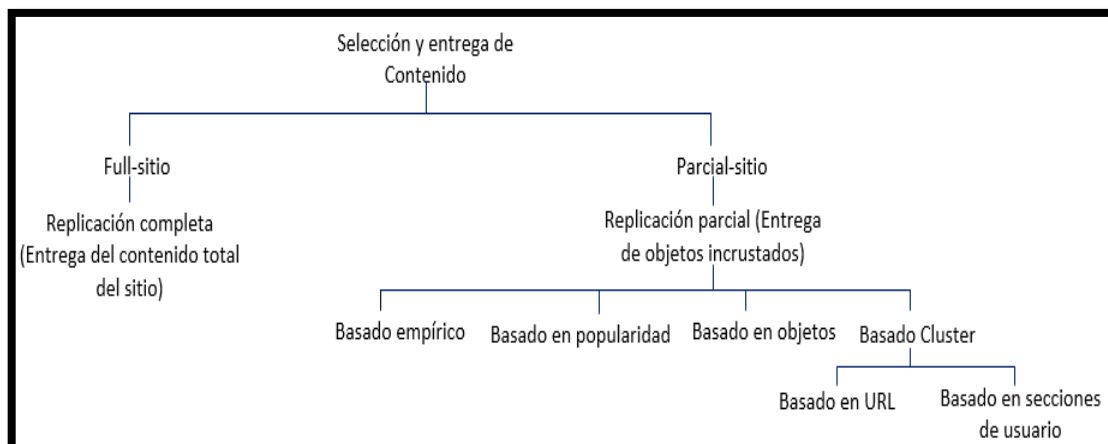


Figura: 2.11. Esquema selección y entrega de contenido [2]

La selección y distribución por el método full-sitio se basa en hacer una replicación total del servidor con el objetivo de entregar la información total al usuario final. De esta manera el encargado de proveer el servicio solo realiza la configuración en su DNS⁶ para que todas las solicitudes del cliente se han resueltas a nivel del servidor CDN. La principal ventaja es que tiene un enfoque muy simple, pero esta solución no resulta tan factible con el aumento de los tamaños de los objetos que ha tenido la Web estos últimos tiempos, y donde evidenciamos el problema es en la actualización de estos objetos que al ser de gran tamaño puede resultar inmanejable.

Por otra parte, la selección y distribución de contenido parcial, los servidores sustitutos realizan una réplica parcial de ciertos objetos del contenido desde el CDN más cercano. De esta manera el proveedor tienen enlaces directos con un objeto específico otorgándole un host el cual es propietario de la CDN. Por lo tanto, el contenido Web recupera ciertas partes del servidor de origen y los objetos que son incrustados se los hace directo del servidor CDN. Este enfoque nos ayuda a reducir la carga al servidor de origen y a la infraestructura CDN.

La selección de contenido va a variar dependiendo de la estrategia que se use para la replicación del contenido Web. El enfoque parcial además de lo ya mencionado presenta una subdivisión en la cual actúa con mayor granularidad.

- ***El basado en el método empírico:*** La persona que gestiona el sitio Web selecciona empíricamente el contenido a ser duplicado por los servidores perimetrales.

⁶ Es una base de datos distribuida usada por aplicaciones TCP/IP para mapear entre nombres de hosts (que vienen dados por una cadena ascii) y direcciones IP (en forma binaria) , también provee a los correos electrónicos información de ruteo.

- **El basado en la *popularidad*:** Este tipo de enfoque se basa en que los objetos más populares son replicados por el servidor. Este tipo de perspectiva no es tan fiable ya que la popularidad del objeto es variable.
- **El basado en el objeto:** El contenido se replica en los servidores sustitutos en unidades de objetos. “Este enfoque resulta ser ambicioso ya que cada objeto se replica en el servidor alternativo (con limitaciones de almacenamiento) que le da la máxima ganancia de rendimiento” [2].
- **El basado en clúster⁷:** En este enfoque el contenido Web es agrupado dependiendo de la frecuencia de visita al sitio o con cualquier tipo de correlación el contenido y así se replica el contenido por grupos. El procedimiento de agrupamiento se realiza ya sea mediante la fijación del número de grupos o mediante la fijación del diámetro máximo de clúster, ya que ni el número ni el diámetro de los grupos nunca pueden ser conocidos. La agrupación puede ser contenido de los usuarios basados en secciones o basado en URL. “Los resultados experimentales muestran que la replicación de contenido basado en este tipo de enfoques de clustering reducir el tiempo de descarga del cliente y la carga en los servidores” [2]. Sin embargo, estos sistemas presenta el inconveniente de dificultad para ser implementada.

2.3.2 Colocación de Sustitutos

La ubicación de los servidores sustitos representa un tema muy interesante al momento que deseamos mejorar la entrega de contenido, y por tal razón que el servidor sustituto tenga una ubicación estratégica. Al tener al sustituto en una ubicación privilegiada el usuario percibirá una calidad de servicio mucho mejor ya

⁷ Se aplica a los conjuntos o conglomerados de computadoras construidos mediante la utilización de hardware común y que se comportan como si fuesen una única computadora.

que la latencia se reduciría y el consumo de ancho de banda sería menor. Para lograr este objetivo se puede determinar dos métricas la cual sería la reducción de costo para el proveedor CDN y la infraestructura de comunicación que necesitamos. Por eso una buena ubicación de los servidores sustitutos permite tener un servicio de calidad y disminuir los costó para CDN. En la Figura: 2.12. mostramos diferentes técnicas e colocación de servidores sustitutos.

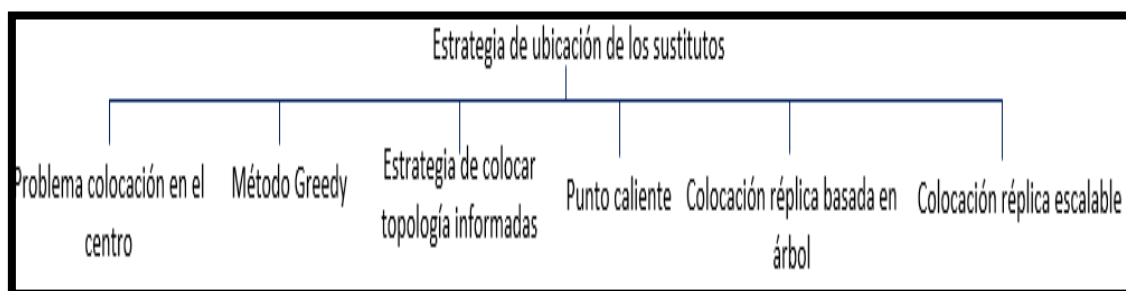


Figura: 2.12. Esquema Estrategia de ubicación de los sustitutos [2]

Teóricamente se enfoca en el problema mínimo k-centro y arboles k-jerárquicamente bien separados (k-HST) para modelar el problema de la colocación del servidor. Para el problema de colocación de centro se define de la siguiente manera: para la colocación de un número determinado de centros, minimizar el la máxima distancia entre un nodo y el centro más cercano. “El algoritmo k-HST resuelve el problema de la colocación del servidor de acuerdo a la teoría de grafos. En este enfoque, la red se representa como un gráfico $G(V, E)$, donde V es el conjunto de nodos y $E \subseteq V \times V$ es el conjunto de enlaces” [2].

El algoritmo consta de dos fases. En la primera fase, un nodo se selecciona arbitrariamente del grafo completo (partición principal) y todos los nodos que están dentro de un radio de azar de este nodo formar una nueva partición (partición secundaria). El radio de la partición hijo es un factor de k menor que el diámetro de la partición padre. Este proceso continúa hasta que cada uno de los

nodos está en una partición propia. Así, el gráfico se repartió de forma recursiva y un árbol de particiones se obtiene con el nodo raíz siendo toda la red y los nodos hoja siendo nodos individuales en la red. En la segunda fase, un nodo virtual se asigna a cada una de las particiones en cada nivel.

Cada nodo virtual en una partición padre se convierte en el padre de los nodos virtuales en las particiones secundarias y juntos los nodos virtuales forman un árbol. “Después, se aplica una estrategia codicioso para encontrar el número de centros necesarios para el árbol k-HST resultado cuando la distancia máxima de centro-nodo está limitada por D . El problema mínimo k-centro puede ser descrito como sigue” [2]:

1. “Dado un grafo $G (V, E)$ con todos sus bordes dispuestos en orden de costo borde c no decreciente: $c(e_1) \leq c(e_2) \leq \dots \leq c(e_m)$, la construcción de un conjunto de gráficos plaza $G_2^1, G_2^2, \dots, G_2^m$. Cada gráfico cuadrado de G , denotado por G_2^i es el gráfico que contiene nodos V y bordes (u, v) siempre que haya un camino entre u y v en G ” [2].
2. “Calcule el máximo conjunto independiente para cada G_2^i . Un conjunto independiente de G_2^i es un conjunto de nodos en G que son al menos tres saltos aparte en G y un conjunto independiente máxima M_i se define como un conjunto independiente V_i de tal forma que todos los nodos en $V_i - V_i$ están a lo sumo un salto de distancia de los nodos en V_i ” [2].
3. “Encuentra más pequeño i tal que $M_i \leq K$, que se define como j ” [2].
4. “Por último, M_j es el conjunto de centro K ” [2].

Debido a la complejidad computacional de estos algoritmos, se han desarrollado algunas técnicas, como la colocación de réplicas Greedy y estrategia de colocación de topología informada. Estos algoritmos subóptimos tienen en cuenta la información existente de CDN, tales como los patrones de carga de trabajo y la topología de la red. Ofrecen soluciones suficientes con costo de computación inferior. El algoritmo Greedy elige entre servidores M de N sitios potenciales. En la primera iteración, el costo asociado con cada sitio se calcula. Se supone que el acceso de todos los clientes converge al sitio bajo. Por lo tanto, se elige el sitio más bajo costo. En la segunda iteración, el algoritmo Greedy busca un segundo sitio (lo que supondría el siguiente precio más bajo) en conjunción con el sitio ya elegido. La iteración continúa hasta que se hayan elegido los servidores M . “El algoritmo voraz funciona bien incluso con datos de entradas imperfectos. Pero se requiere el conocimiento de las ubicaciones de clientes en la red y todas las distancias entre nodos” [2].

En la estrategia de colocación topología informada, los servidores se colocan en los hosts de candidatos en orden de grados descendente. Aquí el supuesto es que los nodos con más grados pueden llegar a más nodos con una latencia menor. “Este enfoque utiliza sistemas autónomos (AS) topologías donde cada nodo representa un solo AS y enlace nodos correspondientes a Border Gateway Protocol (BGP). En este enfoque, cada LAN asociada a un router es un sitio potencial para colocar un servidor, en lugar de cada una como un sitio” [2].

El algoritmo Punto caliente realiza reproducciones cerca de los clientes que generan mayor carga. “Ordena los N posibles sitios según cantidad de tráfico generado por los lugares cercanos y reproducciones en los sitios M principales que generan mayor cantidad de tráfico” [2].

El algoritmo de colocación réplica basada en árbol se basa en la suposición de que las topologías subyacentes son árboles. “Este algoritmo modela el

problema colocación réplica como un problema de programación dinámica. En este enfoque, un árbol T se divide en varios árboles pequeños T_i y la colocación de t proxies se consigue colocando t_i proxies de la mejor manera en cada pequeño árbol T_i , donde $t = \sum_i t_i$ [2].

El algoritmo de réplicas escalable, genera réplicas en la demanda y los organiza en un árbol multicast⁸ a nivel de aplicación. “Este enfoque minimiza el número de réplicas al reunirse con los clientes el servidor genera restricciones de latencia y limitaciones de capacidad” [2].

Para la colocación de servidores replicas el proveedor de CDN debe analizar el número óptimo que se requiere para prestar los multi-ISP e Internet. Para proveer los servicios de Internet el administrador CDN lo más común que haga es desplegar mínimos 40 servidores sustitutos para cumplir con la entrega de contenido. El enfoque de internet se basa básicamente en poner mínimo dos sustitutos en las ciudades más importantes dentro de la cobertura ISP. Tener un ISP para lograr una cobertura de red global sin depender de otros ISP puede resultar viable si no tomamos en cuenta que los sustitutos pueden quedar ubicados en lugares distantes de los clientes de una CDN.

En multi-ISP, el proveedor de CDN coloca numerosos servidores sustitutos en tantos puntos de ISP globales de presencia (POP) como sea posible. “Supera los problemas de enfoque de un solo proveedor de Internet y sustitutos se colocan cerca de los usuarios y el contenido de este modo se entrega de forma fiable y oportuna de los ISP del cliente solicitante” [2].

⁸ Es el envío de la información en múltiples redes a múltiples destinos simultáneamente. Antes del envío de la información, deben establecerse una serie de parámetros.

2.3.4 Externalización del Contenido

Una vez logrado colocar servidores sustitutos en lugares adecuados y determinado el contenido que va hacer entregado, la externalización del contenido resulta una práctica importante, por tal razón se muestran distintos métodos que nos ayudaran con la externalización de contenido.

- **Cooperative push-based:** En este enfoque el contenido se empuja a los servidores sustitutos desde el origen, y servidores sustitutos cooperan para reducir la replicación y el costo de actualización. “En este esquema, la CDN mantiene una correspondencia entre el contenido y servidores sustitutos, y cada solicitud se dirige al servidor alternativo más cercano o de otro modo la solicitud se dirige al servidor de origen. Todavía se considera como un enfoque teórico, ya que no ha sido utilizado por cualquier proveedor de CDN comercial” [2].
- **Non-cooperative pull-based:** En este enfoque las solicitudes de clientes se dirigen a sus servidores sustitutos más cercanos. “Si hay un error de caché, servidores sustitutos tiran contenido del servidor de origen. Muchos CDN utilizan este enfoque, ya que el enfoque basado en el impulso de cooperación se encuentra aún en fase experimental” [2].
- **Cooperative pull-based:** Las solicitudes de los clientes se dirigen a la madre sustituta más cercana a través de la redirección de DNS. “El uso de un índice de distribución, los servidores sustitutos encontrar copias cercanas de contenido solicitado y lo almacenan en la memoria caché. Este enfoque es reactivo en el que un objeto de datos se almacena en caché sólo cuando el cliente lo solicite” [2].

2.3.5 Organización y gestión de la caché

Organización de caché está compuesta por las técnicas de almacenamiento en caché y la frecuencia de actualización de caché para asegurar la disponibilidad y fiabilidad de los contenidos. Aparte de estos dos, la organización de caché también puede implicar el uso integrado de almacenamiento en caché y replicación en la infraestructura de un CDN. Esta integración puede ser útil para un CDN para la gestión de contenidos eficaz.

Almacenamiento en que se realiza en caché de contenido en CDN puede ser intra-cluster o inter-cluster. En una esquema basado en consultas, en un error de caché de un servidor CDN emite una consulta a otros servidores CDN cooperantes. Los problemas con este esquema son el tráfico de consultas significativas y el retraso debido a un servidor CDN tiene que esperar a la última "miss" respuesta de todos los sustitutos cooperantes antes de concluir que ninguno de sus compañeros tiene el contenido solicitado. Debido a estos inconvenientes, el esquema basado en consultas sufre de sobrecarga de aplicación.

El enfoque basado en el resumen supera el problema de las consultas de inundación en el esquema basado en consultas. En el esquema basado en resumen, cada uno de los servidores CDN mantiene un compendio de contenidos en poder de los otros sustitutos cooperantes. Los sustitutos cooperantes son informados acerca de cualquier tipo de actualización del contenido por el servidor CDN de actualización. En el registro de digerir el contenido, un servidor CDN puede tomar la decisión de encaminar una petición de contenido a un sustituto particular. El principal inconveniente es que sufre de sobrecarga de tráfico actualización, debido al cambio frecuente del tráfico de actualización para asegurarse de que los sustitutos cooperantes tienen la información correcta sobre la otra.

El esquema basado en el directorio es una versión centralizada del régimen basado en la digestión. En el esquema basado en directorio, un servidor centralizado mantiene la información de contenido de todos los sustitutos cooperantes dentro de un clúster. Cada servidor CDN sólo notifica al servidor de directorio cuando se producen cambios locales y consulta al servidor de directorio cuando se produzca un error de caché local. Este esquema experimenta cuellos de botella y un punto único de fallo ya que el servidor de directorio recibe actualización y consulta de tráfico de todos los sustitutos cooperantes.

En el esquema basado en hash, los servidores CDN cooperantes mantienen la misma función hash. Un servidor CDN designado tiene un contenido basado en la URL del contenido, las direcciones IP de los servidores CDN, y la función de hash. Todas las solicitudes de que el contenido concreto se dirige a ese servidor designado.

Esquema basado Hash⁹ es más eficiente que otros sistemas ya que cuenta con la más pequeña sobrecarga implementación y mayor eficiencia el intercambio de contenidos. Sin embargo, no escala bien con las solicitudes locales y entrega de contenido multimedia desde las solicitudes de los clientes locales se dirigen hacia y servidos por otros servidores CDN designados. Para superar este problema, un esquema basado semihashing se puede seguir.

Bajo el esquema basado semi-hash, un servidor local CDN asigna una cierta capacidad de su espacio de disco para almacenar en caché el contenido más popular para los usuarios locales y la parte restante de cooperar con otros servidores CDN a través de una función hash. Como hash puro, semi-hash tiene

⁹ Son algoritmos que consiguen crear a partir de una entrada (ya sea un texto, una contraseña o un archivo, por ejemplo) una salida alfanumérica de longitud normalmente fija que representa un resumen de toda la información que se le ha.

pequeña sobrecarga aplicación y contenido de alta eficiencia compartir. Además, se ha encontrado que aumenta significativamente la tasa de éxito local de la CDN.

Un régimen basado en la consulta se puede utilizar para inter-cluster de almacenamiento en caché. En este enfoque, cuando un clúster falla para servir a una petición de contenido, consulta otro grupo vecino. Si el contenido se puede obtener de este vecino, responde con un mensaje de "éxito" o si no, reenvía la petición a otros grupos vecinos. Todos los servidores CDN dentro de un uso del clúster hash esquema basado para servir solicitud de contenido y el servidor CDN representante de un clúster sólo consulta al servidor designado de ese grupo para servir a una solicitud de contenido.

Por lo tanto, este sistema utiliza el esquema basado en hash para el enrutamiento de contenido intra-grupo y el esquema basado en consultas para el enrutamiento de contenido inter-cluster. Este enfoque mejora el rendimiento ya que limita la inundación de tráfico de consultas y soluciona el problema de los retrasos al recuperar contenido desde servidores remotos mediante el uso de un tiempo de espera y TTL con cada mensaje de consulta.

CAPITULO III

TÉCNICAS DE REPLICACIÓN DE CONTENIDO DINÁMICO, ESCALABLE Y EFICIENTE

3.1 Web Caching

El Web Caching es un procedimiento en el cual se realiza un almacenamiento temporal con el fin de que la entrega del contenido web sea más eficiente y veloz al cliente final. En las redes de distribución de contenidos (CDN), diversos servidores realizan copias de contenido como audio, multimedia, imágenes, video y texto estático, en diversos puntos de la red. Las demandas de servicio por parte de los usuarios es receptada por el servidor CDN más cercano, el cual tiene el almacenamiento en caché del contenido, lo que nos ahorra tiempo de carga y reduce el tráfico de la red cliente - servidor.

Hay diferentes clases de cachés de objetos Web:

- **Caché del navegador:** Primero el navegador busca los objetos que se encuentran en la cache antes de solicitarlo a un sitio web. El almacenamiento en caché usualmente usa objetos Web para acelera

navegación. Este almacenamiento generalmente se da con los sitios google.com o yahoo.com si estos están presentes en la barras de navegación podemos deducir que están en la caché de nuestro navegador, el navegador buscará en la memoria caché y cargará inmediatamente el respectivo sitio web.

- **Caché de proxy:** Un proxy caché se ubica cerca de los clientes de la web, en este caso puede ser una empresa. Los usuarios de la empresa deben configurar su navegador web para utilizar el proxy que sea definido, de esta manera los requerimientos solicitados por los usuarios de un sitio web son interceptados y gestionados por el proxy caché. Si no están en la memoria caché, el proxy los soporta en otra memoria caché o desde la propia página web.

- **Caché de proxy transparente:** Al usar un caché proxy común es necesario realizar las configuraciones pertinentes en el navegador de cada usuario. Un proxy caché transparente toma los requerimientos Web sin que el navegador sea consciente de la acción que fue tomada. Proxies transparentes son colocados como elementos intermedios de red antes del usuario final para que todas las solicitudes Web necesariamente pasen a través del proxy.

- **Reversa (inversa) de caché de proxy:** Para disminuir la carga en un sitio web, un caché de proxy usa el proxy reversa y se coloca en frente del servidor para atender las peticiones. Si el proxy inverso no tiene el objeto Web solicitado, se obtiene el objeto de otro caché o desde la propia página web.

Para aplicar el método de web caching en una CDN se han propuesto el push-caching el cual replicar páginas HTML basado en el conocimiento global de la topología de la red y los patrones de acceso de los clientes. Últimamente, se presentan caching adaptativo web y la memoria caché de resumen para permitir el intercambio de cachés Web. El Caché intercambia contenidos periódicamente con otras cachés, eliminando el retraso y el uso innecesario de los recursos de caché. Sin embargo, cada servidor proxy necesita enviar la actualización del índice de contenidos almacenados en caché a todos los demás servidores proxy, y necesita almacenar los índices de contenido de todos los demás servidores proxy.

3.2 CDN extraíble basado en Un-Cooperative

Últimamente CDN se ha dedicado a comercializar servicios tales como contenido de internet, Hosting Web y trasmisión de la entrega de medios. El comportamiento básico de un CDN es tomar el contenido de los servidores de borde y procesar las solicitudes de los clientes. Distintos métodos se ha usado como la rescritura de URL, redirección HTTP, redirección basado en DNS, entre otros han sido propuestos para todas las solicitudes de los clientes de objetos y contenido sean solicitados a los nodos de una CDN. La mayoría de empresas que se dedica a prestar servicios de CDN usa la redirección basado en DNS.

Peticiones de los clientes se dirigen (utilizando ya sea el redireccionamiento de DNS o mecanismos de reescritura de URL) para su servidor sustituto más cercano. Si hay un fallo de caché (es decir, el contenido solicitado no se encuentra), la solicitud se dirige ya sea a un servidor alternativo mirando de la CDN subyacente o al servidor de origen. “Un problema de esta práctica es que CDN no siempre eligen el servidor óptimo desde donde servir el contenido. Además, no es el costo de replicación excesivo, en términos de número de réplicas” [7].

En la Figura: 3.13. mostramos la operación de CDN extraíbles Basado Un-Cooperative, la secuencia de operaciones para un cliente para recuperar una dirección URL. La solicitud de resolución de nombres se envía al servidor de nombres CDN a través del servidor DNS local. Debido a la naturaleza del servicio de localización centralizada, el servidor de nombres CDN no puede permitirse llevar un registro de la ubicación de cada réplica URL. Por lo tanto, sólo puede redirigir la solicitud basada en la proximidad de la red, la disponibilidad de ancho de banda y la carga del servidor. El servidor CDN que recibe la solicitud redirigida puede no tener la réplica. En ese caso, se enviará una réplica del servidor de contenido Web, entonces se responde al cliente.

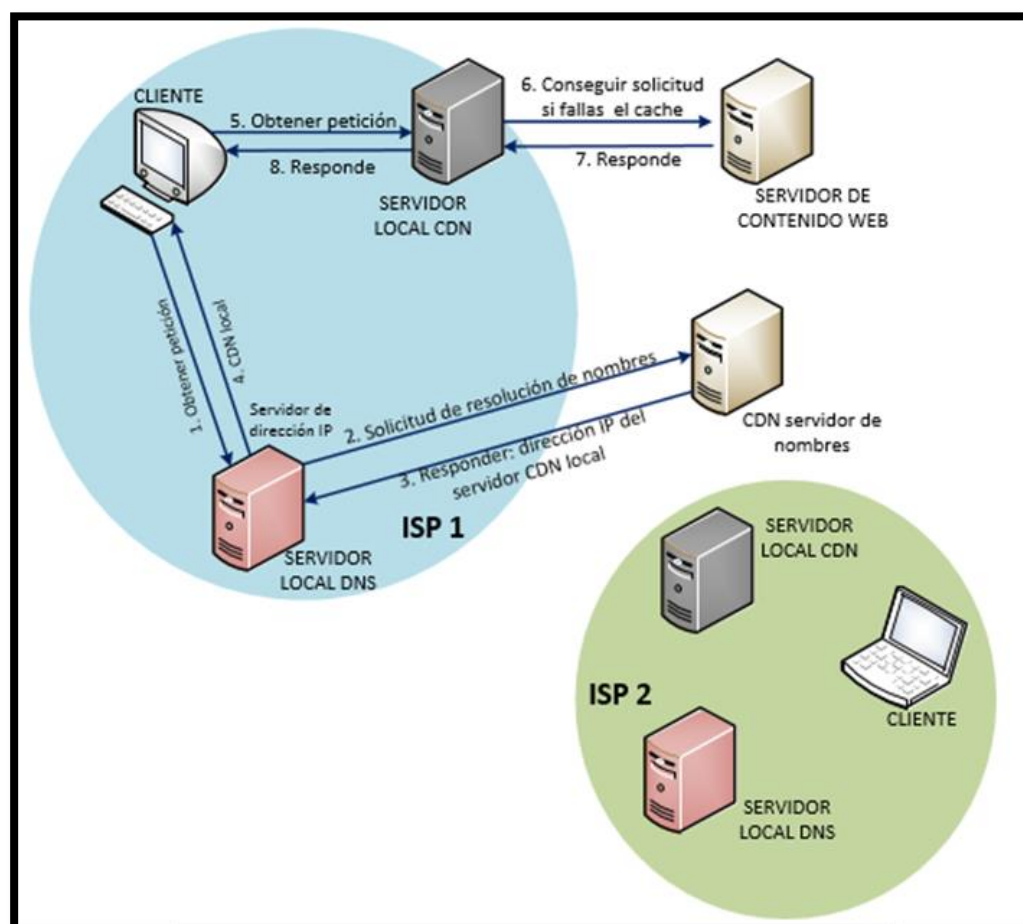


Figura: 3.13. Arquitectura CDN extraíble basado en Un-Cooperative [7]

Actualmente CDN a menudo pone más réplicas de lo necesario y consume gran cantidad de recursos de almacenamiento y actualización.

3.3 CDN insertable basado en Cooperative

La principal ventaja de este esquema es que usamos push en vez de pull, pero proviene del intercambio cooperativo de las réplicas desplegadas. Este intercambio cooperativo reduce significativamente el número de réplicas desplegado y por lo tanto reduce los costos de replicación y actualización.

La arquitectura CDN para este caso es similar a la mostrada en la Figura: 3.14. donde el contenido se empuja (de forma proactiva) desde el servidor Web de origen a Servidores sustitutos CDN. Inicialmente, el contenido es pre-buscado en los servidores sustitutos y, a continuación, los servidores sustitutos cooperar con el fin de reducir el costo de la replicación y actualización. En este esquema, la CDN mantiene una correspondencia entre el contenido y servidores sustitutos, y cada solicitud se dirige al servidor alternativo más cercano. Este servidor puede o no puede tener una réplica del objeto solicitado. Si es así, la petición se sirve localmente, incurriendo en ningún tráfico sobre la red troncal. “De lo contrario, reenvía la solicitud al servidor más cercano que tiene la réplica objeto y transmite la respuesta al cliente. En este caso, el servicio de petición indirecta genera tráfico a través de la columna vertebral de la red entre los dos servidores que participan en la operación” [7].

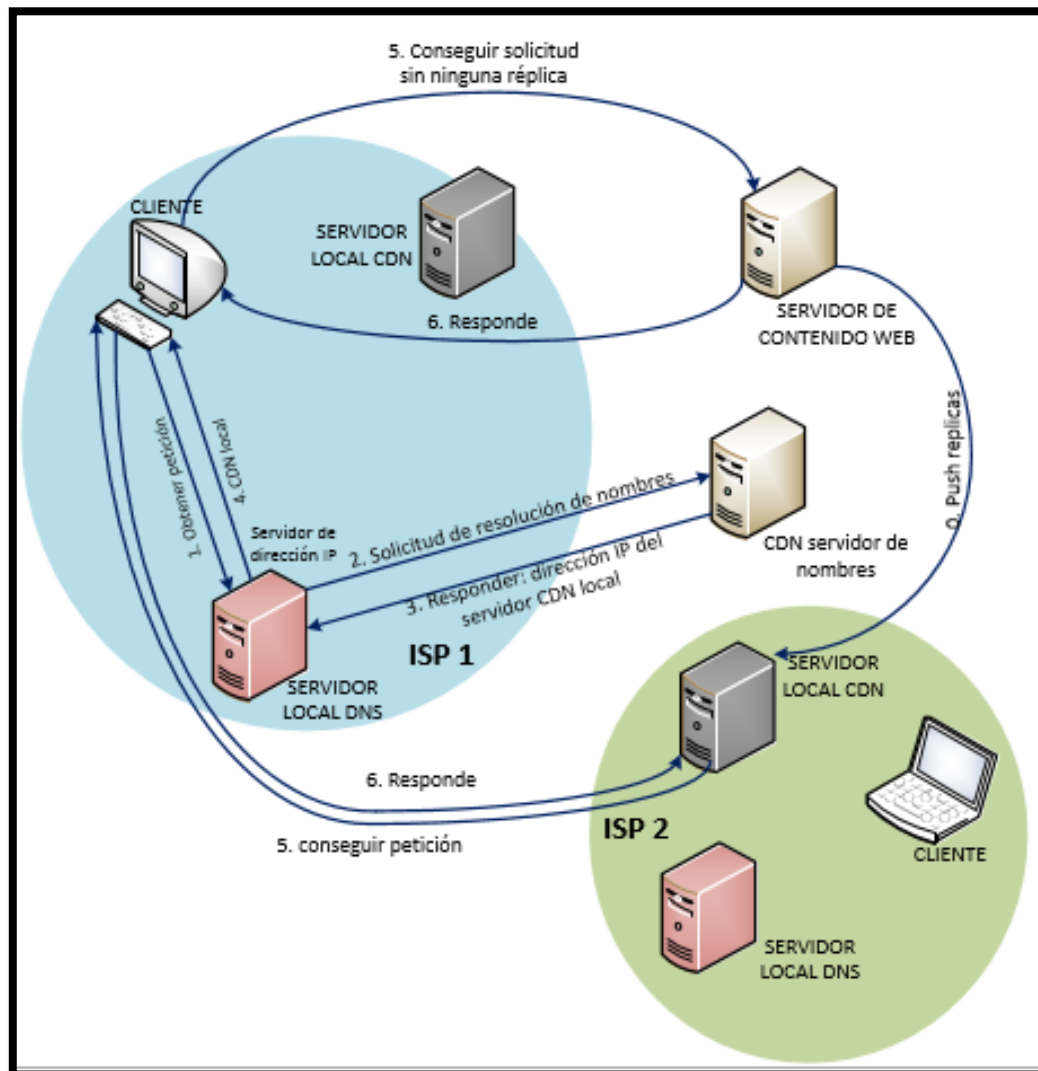


Figura: 3.14. Arquitectura CDN insertable basado en Cooperative [7]

Con tal seguimiento de la ubicación de réplica, el servidor de nombres CDN puede redirigir la petición de un cliente para su réplica más cercana. Se debe considerar que la redirección basado en DNS permite la resolución de direcciones a nivel per-host. Nosotros combinamos con la modificación de contenidos (por ejemplo, la reescritura de URL) para lograr la redirección por objeto. Las referencias a objetos son diferentes reescrito en diferentes nombres de host. Para reducir el tamaño de los espacios de nombres de dominio, los objetos pueden ser agrupados como estudiado, y cada racimo comparte la misma hostname. Desde el proveedor de contenido puede reescribir incrustado URL a priori antes de

empujar los objetos, que no afecta a la latencia percibida de los usuarios y la sobrecarga de una sola vez es aceptable. En ambos modelos, los servidores de borde CDN están autorizados a ejecutar sus algoritmos de sustitución de caché. Es decir, la asignación en la replicación basada en empuje cooperativa es por el estado blando. “Si el cliente no puede encontrar el contenido en el servidor de borde CDN al cual es redirigida su petición, ya sea el cliente podrá pedir al servidor de nombres CDN para otra réplica, o el servidor de borde enviará del contenido del servidor Web y responde al cliente” [8].

3.4 Objeto de sistemas de Localización

El crecimiento de la red de internet y el auge tecnológico que se ha dado en estos últimos tiempos ha permitido que se tenga el acceso a distintas aplicaciones desde cualquier dispositivo electrónico esto involucra un parámetro más a tener en cuenta la movilidad, el hecho de que CDN estén cerca del cliente no nos garantiza que el cliente va a estar en un lugar estático, para lo cual se ha desarrollado ciertos enfoques para dar soporte a este tipo de situaciones. Estos enfoques se pueden en los clasificar siguientes tres grupos: Servicios de directorio centralizado (CDS), Servicios de directorio replicadas (RDS), y servicios de directorio distribuidos (DDS).

3.4.1 Servicios de directorio centralizado (CDS) y Servicios replicadas Directory (RDS)

CDS: Se centra en un único servidor y proporciona información sobre la ubicación de cada objeto. Debido a que reside en un único servidor, es extremadamente vulnerable a los ataques de denegación de servicio. Una variante de esto es el servicio de directorio replicado (RDS), que proporciona múltiples servidores de directorio.

RDS: proporciona una mayor disponibilidad, pero sufre sobrecarga. Aquí no tenemos en cuenta el servicio de directorio con particiones, ya que a menudo requiere servidor de directorio final extra para mantener la información de partición, como el servidor raíz de DNS.

En un CDS los clientes están en contacto con un único directorio para descubrir la ubicación de una réplica cercana. Los clientes posteriormente se ponen en contacto con la réplica directamente. Y RDS ofrece varios directorios, como se lo muestra en la Figura: 3.15.

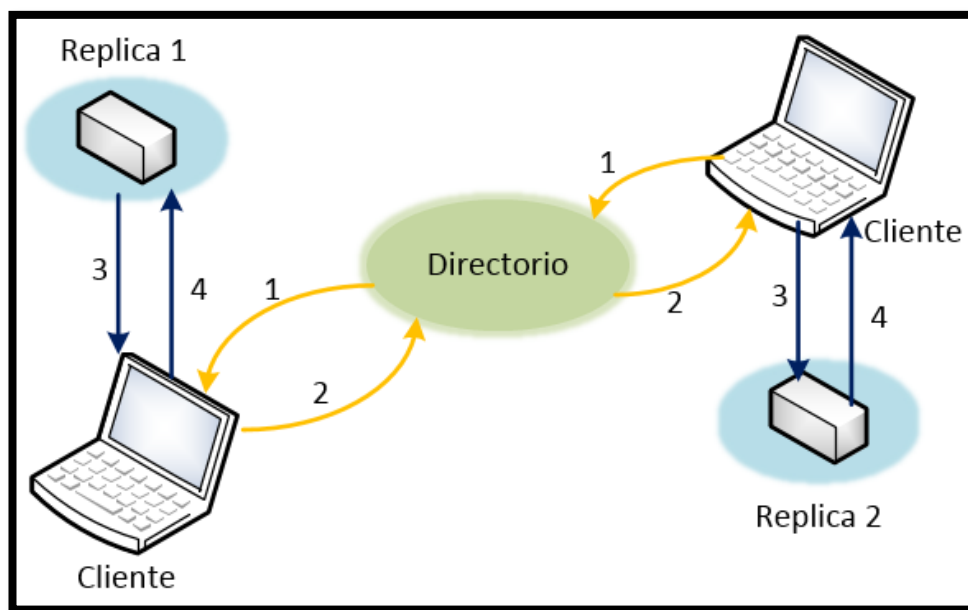


Figura: 3.15. Servicio de directorio centralizado (CDS) y Un servicio replicado Directorio (RDS) [7]

3.4.2 Servicios de directorio distribuidos (DDS)

Gente dedicada al desarrollo de las redes descentralizadas peer-to-peer ha enfocado sus estudios en los diversos métodos como son:

- Las tablas hash distribuidas (DHT) las cuales crean una infraestructura que puede ser utilizada para levantar servicios más complejos, como sistemas de archivos distribuidos sistemas de distribución de contenido, caché web cooperativo, compartición de archivos peer-to-peer, anycast¹⁰, servicios de DNS, multicast y mensajería instantánea.
- El CAN es un protocolo de comunicaciones creado por la marca alemana Robert Bosch GmbH, la cual está basada en una topología bus para la transmisión de mensajes en entornos distribuidos.
- Chord es un protocolo escalable y simple de búsqueda distribuida en redes P2P que relaciona claves con nodos. Está desarrollada para funcionar en redes descentralizadas, y su funcionamiento permite concluir con un resultado exhaustivo de la búsqueda.
- Pastry para implementa un mecanismo de búsqueda es necesario realizar un broadcast a los nodos y eso sería muy costoso. Para solucionar esto lo más factible es usar la aplicación Scribe que permite comunicaciones en grupo mediante Treecasting, es decir, los nodos se organizan a modo de árbol en el que cada elemento es responsable de comunicar el mensaje a sus hijos. Este sistema es muy efectivo para realizar streaming de video o audio.
- Tapestry trabaja de código abierto para el desarrollo de aplicaciones Web Java basadas en componentes y orientadas a objetos. Simplemente, en lugar de tratar con el API Servlet o con las acciones de Struts, el programador de Tapestry almacena los datos de usuario con propiedades de objetos y maneja las acciones de usuario con métodos manejadores de eventos [9].

¹⁰ Es una forma de direccionamiento en la que la información es enrutada al mejor destino desde el punto de vista de la topología de la red.

Estos servicios ofrecen una infraestructura distribuida para localizar objetos de forma rápida con garantía de éxito. En lugar de depender de un solo servidor para localizar un objeto, una consulta en este modelo se pasa alrededor de la red hasta que llega a un nodo que conoce la ubicación del objeto solicitado. La falta de un solo objetivo en los servicios de localización descentralizados significa que proporcionan una disponibilidad muy alta, incluso bajo ataque; los efectos de atacar y desactivación de un conjunto de nodos con éxito se limita a un pequeño conjunto de objetos.

Además, Tapestry es una red superpuesta IP que utiliza una arquitectura tolerante a fallos distribuida para rastrear la ubicación de los objetos en la red. Tapestry tiene dos componentes: una malla de enrutamiento y unos servicios de localización distribuidas.

- ***Tapestry de enrutamiento Mesh:*** En la Figura: 3.16. mostramos que cada nodo se une a la Tapestry de manera distribuida a través de servidores sustitutos cercanos y establecer vínculos vecinos para la conexión con otros nodos del Tapestry. Los enlaces vecinos se muestran como flechas sólidas. Tales enlaces vecinos proporcionan una ruta desde cada nodo a cualquier otro nodo; el proceso de enrutamiento resuelve el destino de dirección de un dígito a la vez [8].

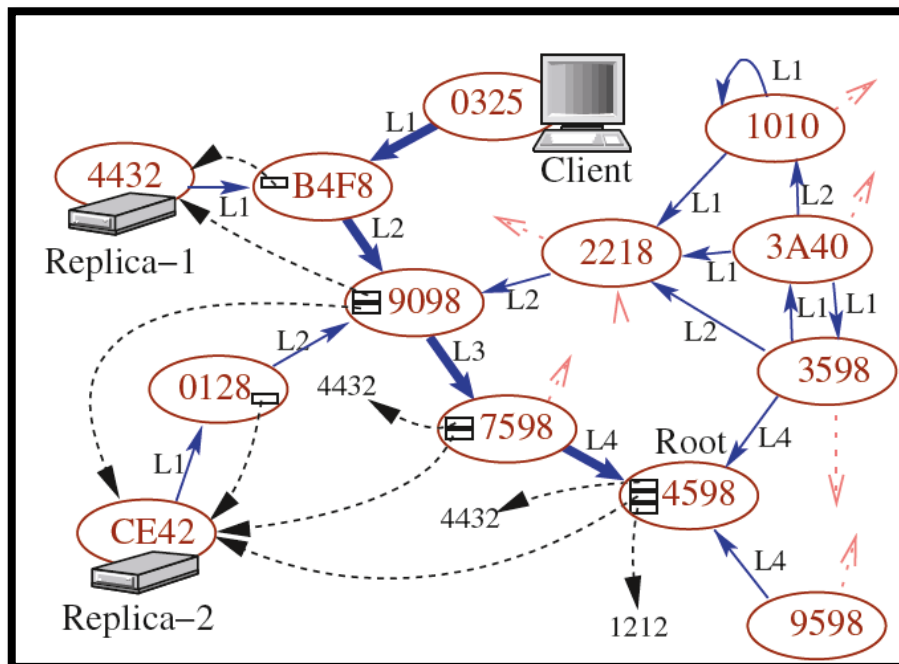


Figura: 3.16. Un directorio distribuido (Tapestry) [2]

- **Distribución de Ubicación de Servicio Tapestry:** Asigna un nombre único global (GUID) a cada objeto. A continuación, asigna determinísticamente cada GUID de un nodo raíz única. Servidores de almacenamiento publica objetos mediante el envío de mensajes hacia las raíces, depositando punteros ubicación en cada salto. Figura: 3.16. muestra dos réplicas y la raíz de la tapicería de un objeto. “Estas asignaciones son simplemente punteros a los servidores aquí se almacena objeto, y no una copia del objeto en sí. Así, para los objetos cercanos, los mensajes de búsqueda de clientes cruzan rápidamente el camino tomado por publicar mensajes, lo que resulta en requerimientos de búsqueda rápidos que explotan localidad. Se muestra que la distancia media recorrida en la localización de un objeto es proporcional a la distancia desde ese objeto” [8].

3.5 Multidifusión para difundir actualizaciones

El Protocolo de Difusión Multicast (MDP) es un protocolo de comunicaciones para uno-a-muchos transmisiones en redes alámbricas o inalámbricas. Protocolos de multidifusión se utilizan para aplicaciones tales como conferencias de grupos o juegos en línea. La primera aplicación MDP, en 1993, se basa en la imagen multicast (IMM), un protocolo utilizado para la difusión de archivos de imágenes de satélite (como imágenes relacionadas con el clima) sobre el Mbone (la parte de Internet que puede manejar la multidifusión IP). “El marco MDP, desarrollado entre 1995 - 1997, fue utilizado en redes multicast vía satélite, redes de simulación de alta velocidad de DARPA y la de todo el mundo Mbone. MDP versión 2, lanzado en 1997, incluye el kit de desarrollo de protocolo marco y el software (SDK), y cuenta con las técnicas más avanzadas de protocolo” [10].

Para que la Multidifusión pueda difundir actualizaciones en IP presenta algunos problemas como la base arquitectónica para la distribución por Internet. “Por ejemplo, sólo funciona a través del espacio, no a través del tiempo, mientras que la mayor distribución de contenidos en el trabajo de Internet a través de ambos. Además, no hay ampliamente disponibles inter-dominio de multidifusión IP” [8].

Como una solución a la multidifusión es apuntar a grupos pequeños con solicitudes de múltiples fuentes, tales como videoconferencia, mientras que otros se centran en gran escala, aplicaciones de una sola fuente, como el streaming de medios de comunicación multidifusión. “Utiliza el servicio de localización de Tapestry para encontrar la raíz de multidifusión, y luego utiliza el enrutamiento de Tapestry de ruta tanto en el control y mensajes de datos. En contraste, sólo usamos el mecanismo de ubicación Tapestry para encontrar la réplica cercana” [8].

CAPITULO IV

ENTREGA Y GESTIÓN DE CONTENIDOS

4.1 Sistemas para la entrega Contenido Web

La Web se ha desarrollado en los últimos tiempos como un medio de distribución de contenido con interés limitado a un medio de comunicación importante, donde el contenido es entregado a los usuarios. Cuando se requiere alta disponibilidad, escalabilidad y rendimiento, una solución común para los proveedores de contenido es explotar los servicios de terceros para mejorar el rendimiento de los contenidos y la prestación de servicios.

El objetivo de los proveedores CDN es garantizar el rendimiento de entrega adecuada incluso cuando la entrante solicitud de carga es abrumador para el proveedor de contenido. “CDN se han propuesto originalmente para distribuir principalmente contenido Web estático y algo de contenido de audio / video streaming limitado a través de Internet” [11].

“CDN en sus inicios fueron diseñados principalmente para aliviar la congestión de la red causada por la entrega de páginas Web estáticas a través de los puntos de interconexión públicos congestionados” [11]. Sin embargo, el

contexto actual de distribución de contenidos es muy diferente al de la creación de estas infraestructuras, que se remontan a hace casi 10 años. De hecho, el escenario web se caracteriza actualmente por una mayor sofisticación y complejidad en el contenido entregado.

“Tecnologías de administración de contenido tradicionales diseñados para el contenido estático no son capaces de responder a las nuevas necesidades, ya que hay retos y limitaciones inherentes en la entrega de contenido dinámico que hay que superar” [11]. En los últimos años, las CDN comenzaron a evolucionar hacia el apoyo para la entrega de contenido generado dinámicamente, lo que permite que los proveedores de contenido puedan explotar los beneficios de la CDN para aplicaciones y servicios web modernos.

A continuación se detallará los sistemas empleados para la generación y entregar de contenido Web que pueden ser empleadas por CDN para acelerar la generación y entrega dinámica de contenido personalizado.

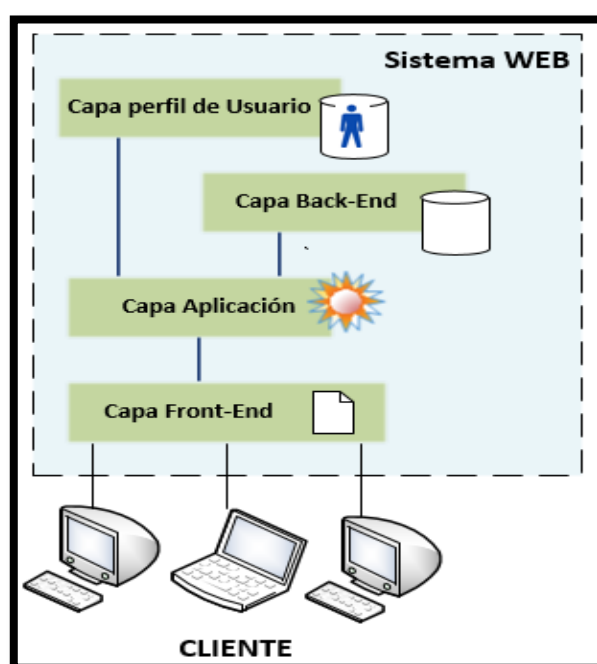


Figura: 4.17. Capas lógicas de un sistema de Web [11]

4.1.1 Capas lógicas de un sistema Web

La gran mayoría de los sistemas de web actuales están basados en una arquitectura lógica con varios niveles que separa la interfaz HTTP, la lógica de aplicación, el repositorio de datos y la información relacionada con el usuario para la autenticación y la personalización de contenido. “Estas capas de la arquitectura lógica se refieren a menudo como front-end, la aplicación de back-end, y capas de perfil de usuario. En la Figura: 4.17. muestra la estructura de un sistema típico de la prestación de servicios basados en la Web” [11].

La capa de aplicaciones para el usuario es la interfaz del servicio basado en la Web. Se acepta solicitudes de conexión HTTP de los clientes, sirve contenido estático del sistema de archivos, y representa una interfaz hacia la capa lógica de la aplicación. La entrega de contenido estático es una operación sencilla, un contenido web estático normalmente se almacena en un sistema de archivos y solicitudes de los clientes de este tipo de contenido son administradas por el servidor HTTP, que recupera los recursos del sistema de archivos y los envía de vuelta al cliente en las respuestas HTTP.

Los contenidos estáticos que son manejados por la capa de aplicaciones hacia el cliente son:

- **Objetos Web integrados en una página Web:** Son objetos comunes en una página Web como imágenes, hojas de estilo, y los componentes activos, tales como animaciones flash, applets Java y controles ActiveX.

- **Contenido Multimedia:** Hace referencia a el audio y secuencias de vídeo que son contenidos estáticos manejados por la capa de aplicaciones para el usuario.
- **Fragmentos de páginas:** Conforman parte de una página Web con un tema con funcionalidad específica. Cada fragmento se considera como una entidad independiente de información distinta. Por ejemplo, la página de un portal Web se describe típicamente como compuesta por fragmentos como novedades, artículos de fondo, barras de enlace y anuncios. “El uso de fragmentos en la gestión de contenido estático tiene como objetivo mejorar la reutilización de contenido Web, debido a que algunos fragmentos son comunes a varias páginas” [11].

La capa de aplicación conforma la parte central de un servicio basado en la Web ya que se encarga de toda la lógica de negocio y calcula la información que se utiliza para construir las respuestas con contenido generado dinámicamente la generación de contenido a menudo requiere la interacción con las capas de fondo y de perfil de usuario. Por lo tanto, la capa de aplicación debe ser capaz de interactuar con la lógica de la aplicación del almacenamiento de datos del programa de fondo y debe poder acceder al perfil de usuario cuando necesita contenido personalizado. Ejemplos de contenidos dinámicos generados por la capa de aplicación son:

- Respuestas recuperadas una fuente organizada de información, como la página de la canasta de compras o búsquedas en un sitio de comercio electrónico.
- Contenido Web se genera dinámicamente para separar el contenido de su representación. Por ejemplo, las tecnologías basadas en sistemas XML de gestión de contenido o proporcionan mecanismos para

separar la estructura y de representación de datos de un documento Web. “En estos sistemas, el contenido se genera dinámicamente a partir de una plantilla sobre la marcha. la carga de la generación de los datos dinámica, que requiere un esfuerzo computacional debido a la recuperación de datos de bases de datos, (opcional) procesamiento de la información y la construcción de la salida de HTTP, se sobrepasaban por la conveniencia de manipulación de los datos a través de software diseñado específicamente para este fin, tal como un DBMS” [11].

- Contenido Web generada por el comportamiento social del usuario. Por ejemplo, las páginas de foros o blogs proporcionan un lugar de intercambio de mensajes escritos por los usuarios de la Web.

La capa de fondo gestiona el principal repositorio de información de un servicio basado en la Web. Por lo general consiste en un servidor de base de datos y almacenamiento de información crítica que es una fuente de generación de contenido dinámico. Si nos referimos a los ejemplos de generación de contenido dinámico de la capa de aplicación, podemos identificar los siguientes repositorios de datos:

- En el caso de un sitio de comercio electrónico, se utiliza una base de datos para almacenar las listas de productos, con acceso para buscar en el catálogo de productos. Por otra parte, también las interacciones de los usuarios se gestionan mediante una base de datos para la cesta de la compra estado o una lista de compras.
- En el caso de un sistema de gestión de contenidos, la generación dinámica de contenido accede a la base de datos para recuperar tanto las plantillas de las páginas web y los contenidos reales durante la generación de recursos Web.

- Para los sitios web como blogs o foros, artículos, comentarios y mensajes normalmente se almacenan en una base de datos

El perfil de usuario tiene la información sobre las preferencias del usuario y el contexto. Se accede a esta información en la generación de contenido dinámico para proporcionar contenido personalizado. La información almacenada en el perfil de usuario puede originarse a partir de múltiples fuentes, tales como:

- Información proporcionada por el usuario es aquella que generalmente a través de un relleno en forma de añadir / editar se sabe las preferencias del usuario. Esta comunicación perfil puede ocurrir cuando el usuario se registra para el acceso a un servicio basado en la Web o se puede llenar / modificarse luego.
- Información inferida a partir del análisis del comportamiento del usuario que normalmente se obtiene de la extracción de datos de los registros Web. Ejemplos típicos de servicios basados en la Web que se basan en la información obtenida a través de la minería de datos son los sistemas de recomendación para el comercio electrónico o la publicidad adaptada a las preferencias del usuario.

4.1.2 Una simplificada CDN Arquitectura

La arquitectura de un CDN tiene como objetivo lograr un alto rendimiento y escalabilidad, aprovechando el principio de replicar los recursos del sistema para garantizar un alto nivel de rendimiento en el servicio de una enorme cantidad de contenido. La replicación se la efectúa tanto a nivel local y distintas localidades geográficas. En el caso de replicación local los recursos del sistema se usan servidores para que el servicio de solicitudes de los usuarios esté estrechamente

conectado. Son colocados en una misma LAN y generalmente se comparten un solo enlace upstream conectando el sistema con resto de la Internet.

El término común para describir un sistema de este tipo es clúster. Servidores dentro de un grupo proporcionan una mayor potencia de cálculo gracias a la replicación de los recursos del sistema. Por otra parte, la replicación puede mejorar la tolerancia a fallos, porque un nodo defectuoso puede ser evitado fácilmente.

Sistemas basados en LAN tienen muchas ventajas, pero tienen problemas de escalabilidad relacionados con la generación y la entrega de recursos eficiente cuando el sitio web es muy popular. “El primer problema que afecta a la replicación a escala local es la llamada primera milla, es decir, el enlace de red que conecta el clúster a Internet. Este enlace puede representar el cuello de botella del sistema para el funcionamiento de extremo a extremo; además, es un potencial punto de falla” [11].

El tráfico en la zona de clúster basado en Web, las fallas en un router externo, y de denegación de servicio (DoS) puede hacer que el servicio sea inalcanzable, independientemente de la potencia de cálculo de la plataforma de clúster. Cuando se necesitan una mejor escalabilidad y rendimiento, es útil para replicar algunos elementos de la infraestructura en una escala geográfica.

Un esquema de la arquitectura distribuida geográficamente de un CDN a la cual nos referimos se muestra en la Figura: 4.18., donde distinguimos dos tipos de servidores en un CDN típico con los servidores de borde y el servidor central. Los servidores Edge se replican en el llamado borde de la red, lo que significa que estos servidores están tan cerca como sea posible a los clientes, por lo general en los puntos de presencia de múltiples proveedores de servicios de Internet (ISP), y

son los principales responsables de la interacción con los clientes. Las solicitudes de cliente son típicamente desviados desde el servidor de origen al servidor de borde mediante el redireccionamiento al DNS-based. Este enfoque se basa en los servidores DNS modificados que resuelven las consultas para el nombre de host sitio con la dirección IP de un servidor perimetral adecuado.

“El servidor central es una entidad lógica que se encarga de las funciones que están relacionadas con la gestión de la infraestructura, la coordinación de las políticas de distribución de la solicitud, y la facturación. Puede ser implementado como un único servidor de gran alcance o, más a menudo, como un multi-cluster, es decir un conjunto de agrupaciones que cooperan y se comportan como un solo equipo virtual con alta disponibilidad y capacidad de cálculo” [11].

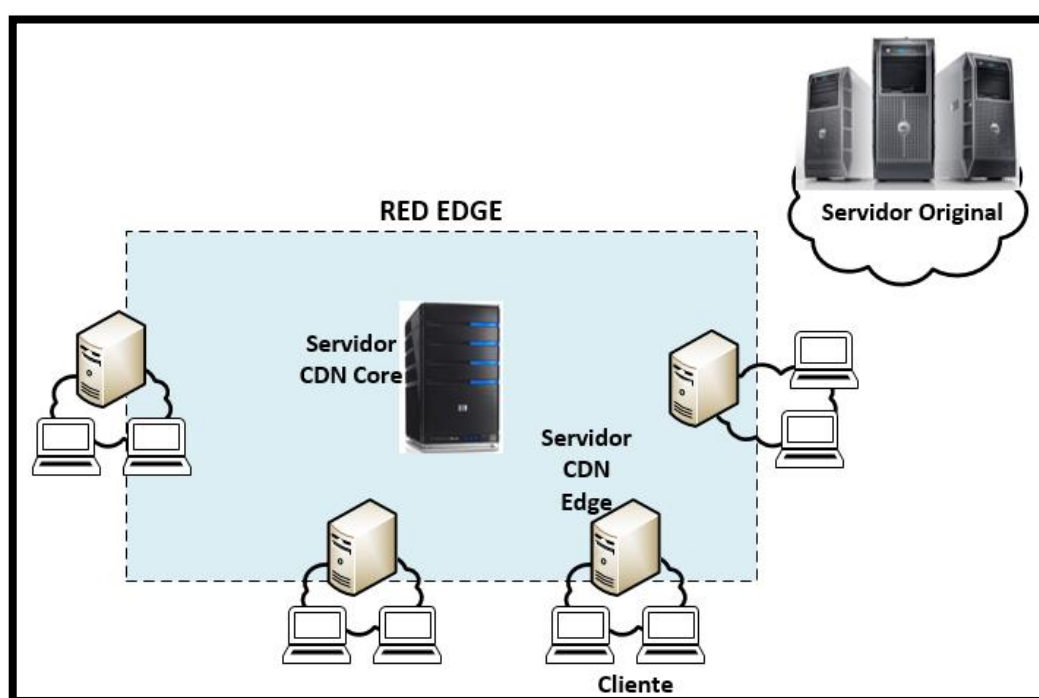


Figura: 4.18. Arquitectura CDN simplificada [11]

4.1.3 Aceleración de la generación de contenidos y Entrega

La exigencia que tiene hoy la Web hacia una mayor demanda de entrega de contenido de alto rendimiento y escalable requiere que el proveedor de contenidos se vea en la necesidad de depender de una CDN. Para esto una CDN debe contar con técnicas para acelerar la entrega de contenido.

Para saber cómo un CDN puede acelerar la entrega de contenido Web y aplicaciones, nos enfocáramos en los principales elementos como el servidor de origen y en los servidores de última generación, que son los elementos de la infraestructura Web más involucrados en el proceso de entrega de contenido. Para mejorar el rendimiento y escalabilidad nos centramos en dos ítem básicamente: el almacenamiento en caché y replicación. CDN replicar algunas capas lógicas del servidor de origen en sus servidores de última generación como lo presentamos en la Figura: 4.19.

- **Capa Front-End:** El servidor perimetral es responsable sólo de la gestión de contenido estático. Este enfoque es típico de la primera generación de CDN, donde los servidores de borde, llamados servidores sustitutos, se comportan como proxies inversos para acelerar la entrega de contenido que se puede almacenar en el nivel de sistema de archivos.
- **Capa Aplicación:** Un CDN se utiliza para mejorar el rendimiento de la entrega de contenido generado dinámicamente. Se mueve programas o componentes de aplicaciones web directamente en el servidor Edge con el objetivo de generar contenido Web dinámico cerca de los clientes.
- **Capa Back-End:** El servidor Edge proporciona tanto las funciones para generar contenido dinámico y anfitriones datos involucrados en la generación de contenidos. El servidor central de la CDN es sólo

responsable de la gestión de la infraestructura y actúa como una copia maestra de los datos.

- **Capa de perfil de usuario:** Los anfitriones del servidor Edge también usan el repositorio de datos utilizados para la generación de contenidos personalizado.

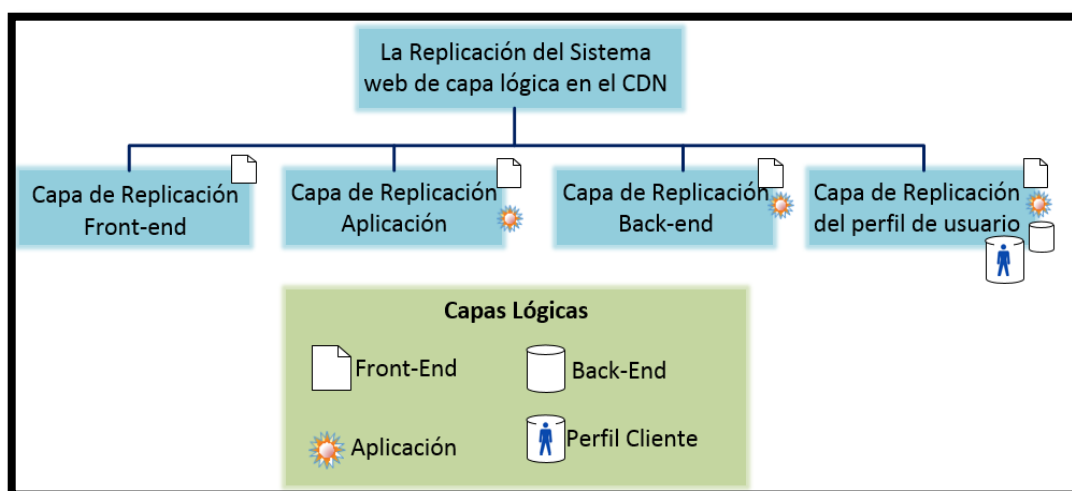


Figura: 4.19. Posible mapeo de sistemas Web capas lógicas en una infraestructura CDN [11]

4.2 Capa de Replicación Front-End

El objetivo de la capa de replicación Front-End es mejorar el rendimiento y la estabilidad en la entrega de contenido estático como se presenta en la Figura: 4.20. Este contenido es almacenado en la caché de los servidores de borde de la CDN.

Al acelerar la entrega de contenido estático utilizando una infraestructura de terceros nos permite mejorar el rendimiento de la entrega de contenido, y se remonta a la primera generación de CDN. Sin embargo, la entrega de este contenido sigue siendo una tarea crítica, debido a la creciente cantidad de contenido multimedia que se está convirtiendo en una fracción significativa de

tráfico Web. Trasladando la entrega de cierto contenido de medios cerca de los clientes puede tener un beneficio importante, debido al gran tamaño de estos contenidos, los retrasos relacionados con la red en los puntos de interconexión pueden tener un impacto significativo en el rendimiento percibida por el usuario y debido a las técnicas de streaming de HTTP, que se utiliza comúnmente, la reducción de la varianza en los resultados de tiempo de entrega es más leve en la reproducción.

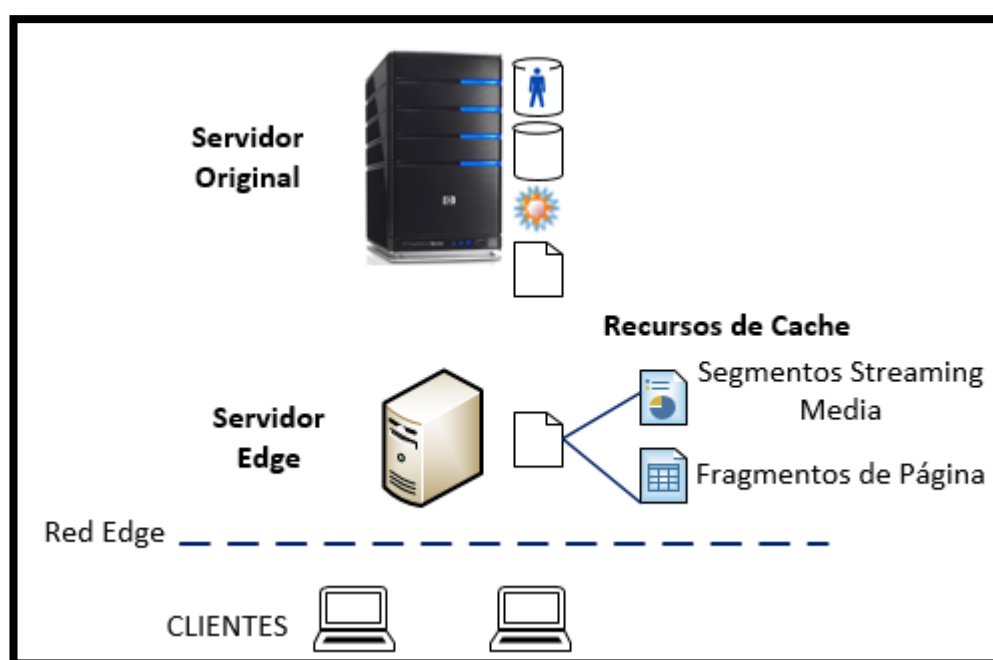


Figura: 4.20. La réplica de la capa de aplicaciones para el usuario en el servidor Edge [11]

Debido al gran tamaño de los contenidos multimedia, que es común a la memoria caché en el servidor de borde sólo la fracciona el contenido más popular de cada segmento multimedia en lugar de almacenar la totalidad del recurso, como se muestra en la Figura: 4.20. La popularidad de cada fragmento va a depender de la preferencia que decida el usuario. “En el caso de acceso secuencial, el enfoque común es confiar en el almacenamiento en caché secuencial, que almacena la primera parte del recurso para reducir el tiempo de almacenamiento en búfer. Por otro lado, cuando los patrones de acceso implican una cantidad significativa de operación de búsqueda dentro de los medios de

comunicación, diferentes técnicas de caché, tales como el almacenamiento en caché intercalada, pueden ser más eficaces” [11].

Se ha propuesto el enfoque de dividir el contenido en streaming en segmentos también para los recursos Web, en el caso de la entrega de contenido Web ensamblado a partir de fragmentos. Esta solución requiere más esfuerzo por parte del servidor de borde, ya que la capa de aplicaciones para usuario debe incluir las funciones para el almacenamiento en caché independiente y el montaje de fragmentos. Al ser una entidad de información independiente, cada fragmento puede tener su propio perfil de caches, que describe la capacidad de almacenar en caché y su Time-To-Live (TTL)¹¹, lo que permite gestionar la actualización del contenido y su trayectoria en un fragmento de granularidad más que en el nivel de la página web [11].

Una vez que el usuario solicita la página Web, el servidor de borde examina su caché con los fragmentos de página incluidos y va uniendo la página mientras se está cargando. Sólo aquellos fragmentos que se consideren imposible para el caché se recuperan desde el servidor de origen. Por lo tanto, se usa el caché basado en fragmentos y montaje dinámico en los servidores de borde, con esto el servidor origen no tiene que montar la página y requiere para entregar sólo una pequeña fracción de la página, correspondiente a fragmentos no puedan colocarse antes en memoria.

En cuanto a la actuación percibida por el usuario, el almacenamiento en caché basada en fragmento se ha demostrado ser eficaz en la mejora del tiempo de respuesta al servir la mayor parte de los recursos que comprenden una página generada dinámicamente en el borde de la Internet, cerca del usuario final . Además, el almacenamiento en caché basado en fragmento tiene también efectos

¹¹ Es un concepto usado en redes de computadores para indicar por cuántos nodos puede pasar un paquete antes de ser descartado por la red o devuelto a su origen.

beneficiosos sobre los servidores de borde. De hecho, mejora la utilización de espacio en disco, ya fragmentos que se comparten a través de diferentes páginas web deben ser almacenados sólo una vez. “Además, reduce la cantidad de invalidación en el servidor perimetral, ya que sólo las partes de la página Web que expire necesitan ser invalidado” [11].

La norma común para el almacenamiento en caché basado en fragmento está representado por Edge Side Includes, que es un lenguaje basado en XML que permite distinguir a través cacheables contenido XML y no almacenable en caché. Algunos diseños de proveedores de contenidos desarrollaron la lógica de negocio para formar y armar las páginas utilizando la especificación ESI dentro de su entorno de desarrollo. Además de la funcionalidad principal para la inclusión de fragmentos dentro de una página, las otras funcionalidades claves proporcionadas por ESI incluyen el apoyo para el manejo de excepciones debido a los fragmentos de indisponibilidad y el soporte de invalidación explícita de fragmentos en caché en una manera tal que garantiza una consistencia más fuerte que la proporcionada por un mecanismo basado en TTL.

Este sistema es capaz de construir objetos complejos a partir de fragmentos y se ha desarrollado para manejar grandes eventos deportivos en los sitios web alojados por IBM. Los autores también abordan el problema de la detección y la actualización de todas las páginas Web afectados por uno o más cambios de fragmentos. La propuesta es adoptar diferentes algoritmos basados en el gráfico de recorrido que se pueden utilizar de acuerdo con los requisitos de coherencia.

La mayoría de los servidores de borde que admiten caché basado en el fragmento no proporcionan ningún apoyo a la cooperación entre los cachés borde individuales, es decir, estos son tratados como entidades totalmente independientes. Esta limitación no permite sacar el máximo provecho de las

capacidades potenciales de almacenamiento en caché de los servidores de borde que pueden ser explotadas mediante la cooperación.

Los principales inconvenientes de la solución basada en el fragmento están relacionados con su aplicabilidad con respecto al tipo de contenido dinámico de ser entregado y a la tarea de fragmentación de una página Web. De hecho, el almacenamiento en caché basada en el fragmento se puede aplicar eficazmente si la corriente de solicitudes se caracteriza por su localidad y si las actualizaciones en el contenido del servidor de origen no son frecuentes.

Esta condición garantiza que los perfiles fragmento de cacheabilidad sean suficientes para la gestión eficiente del contenido, de este modo se puede quitar el peso al servidor de origen quitándole la tarea para invalidar explícitamente fragmentos almacenados en caché. Además, esta técnica sufre de falta de transparencia, ya que el almacenamiento en caché, la fragmentación y el montaje deben aplicarse en función de cada aplicación.

4.3 Capa de Replicación de Aplicación

Un atasco en el rendimiento sucede en CDN cuando replican solamente la capa front-end que está representada por la capa de aplicación en el servidor de origen, quien es el responsable de la generación de contenido dinámico de acuerdo con la lógica de la aplicación Web. La réplica de la capa de aplicación, comúnmente conocido como el filo de computación, tiene como objetivo mejorar el suministro de contenido generado dinámicamente mediante la descarga de esta tarea desde el origen del servidor. El código de la aplicación se replica en varios servidores de última generación, mientras que la capa de datos todavía se mantiene centralizada. El cálculo es empujado hasta el borde de la red representado en la Figura: 4.21.

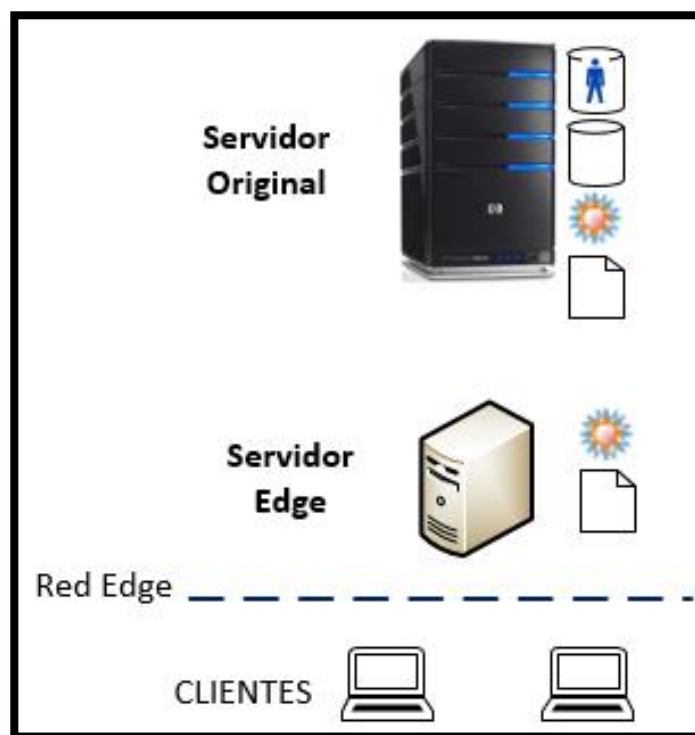


Figura: 4.21. La réplica de la capa de aplicación en el servidor Edge [11]

En informática borde, cada servidor de borde tiene una copia completa del código de la aplicación, mientras que la capa de fondo todavía está centralizada en el servidor de origen, es decir, los servidores de borde siguen compartiendo una base de datos centralizada. Al empujar la tarea de cálculo hasta el borde de la red, la carga en el servidor de origen se puede reducir y el CDN puede lograr una mejor eficiencia y rendimiento y una mayor disponibilidad con respecto a la parte delantera de extremo único enfoque de replicación, donde la aplicación y capas de datos son gestionado de forma centralizada. “Se pueden mencionar dos soluciones arquitectónicas en función de la capacidad del servidor de borde como peticiones transaccionales y no transaccionales” [11].

Una petición transaccional es un conjunto atómico de operaciones de base de datos que suele implicar bloqueo en parte de la base de datos y realizar alguna actualización de los registros de base de datos, mientras que las solicitudes no

transaccionales tienen un comportamiento de sólo lectura de los datos. Si el servidor de borde no puede distinguir el tipo de solicitud, el servidor Web en el servidor reenvía todas las peticiones a su capa de aplicación local, en el que se ejecutan. La lógica de la aplicación local, entonces hace que las convocatorias de acceso a la base de la capa de datos centralizada ubicada en el núcleo CDN.

De lo contrario, si el servidor de borde es capaz de distinguir entre las peticiones transaccionales y no transaccionales, el servidor de borde redirige sólo las solicitudes no transaccionales a la capa de aplicación local, mientras que las solicitudes transaccionales se transmitirán directamente a la capa de aplicación en el servidor de origen, que a continuación, ejecuta la transacción y accede a la base de datos centralizada.

En el objetivo de la replicación de aplicaciones, el núcleo CDN normalmente juega un papel coordinador, encargándose de la migración y / o replicar las aplicaciones en los servidores de borde y hacer el seguimiento de las réplicas. Puede ser también responsable del mantenimiento de las réplicas de una aplicación conforme a la copia principal. El núcleo CDN puede lograr esta funcionalidad utilizando una simple invalidación basada en servidor que es, la actualización de la aplicación en los servidores de borde cuando el desarrollador cambia la copia principal.

Las cuestiones de migración Proceso se han abordado por muchos años de investigación; un ejemplo de sistema complejo que podría ser empleado en el contexto CDN es pmatrix, que migra todo el estado dinámico de la aplicación desde un servidor a otro. Sin embargo, las aplicaciones Web no requieren una migración de aplicaciones reales en un instante arbitrario pero sólo a petición de los límites. Por lo tanto, una importante simplificación aplicable en el contexto CDN es el despliegue automático de la aplicación en los servidores de borde.

La replicación de capa de aplicación se caracteriza para permitir la personalización de las aplicaciones concretas y específicas. El enfoque de la replicación aplicación no es ni genérico ni transparente. De hecho, se requiere la personalización en función de cada aplicación, ya que se necesita una configuración manual para elegir los componentes que se van descargadas y dónde desplegar aplicaciones.

Otras desventajas de la capa de aplicación se enfocan en la replicación para mantener los datos centralizados en el servidor de origen. Esta elección de la arquitectura determina dos inconvenientes. En primer lugar, si los servidores de borde se encuentran en todo el mundo, como en gran escala CDN, entonces cada acceso a datos incurre en un retraso WAN; en segundo lugar, la base de datos centralizada puede convertirse rápidamente en un cuello de botella, ya que tiene que servir a todos de base de datos solicitudes de todo el sistema. “Por lo tanto, la solución de replicación de aplicación es adecuada sólo para los sitios web que requieren pocos accesos a la base de datos centralizada con el fin de generar el contenido solicitado” [11].

4.4 Capa de Replicación Back-End

Con la informática de borde puede no se puede resolver todos los problemas de escalabilidad, ya que en algunas aplicaciones Web el cuello de botella se encuentra en la capa de fondo en lugar de la capa de aplicación. En este caso, los problemas de escalabilidad pueden abordarse mediante la asignación de una CDN para la gestión de datos de la aplicación. Un CDN proporciona respuestas a las preguntas de la capa de aplicaciones alojadas por los servidores de borde.

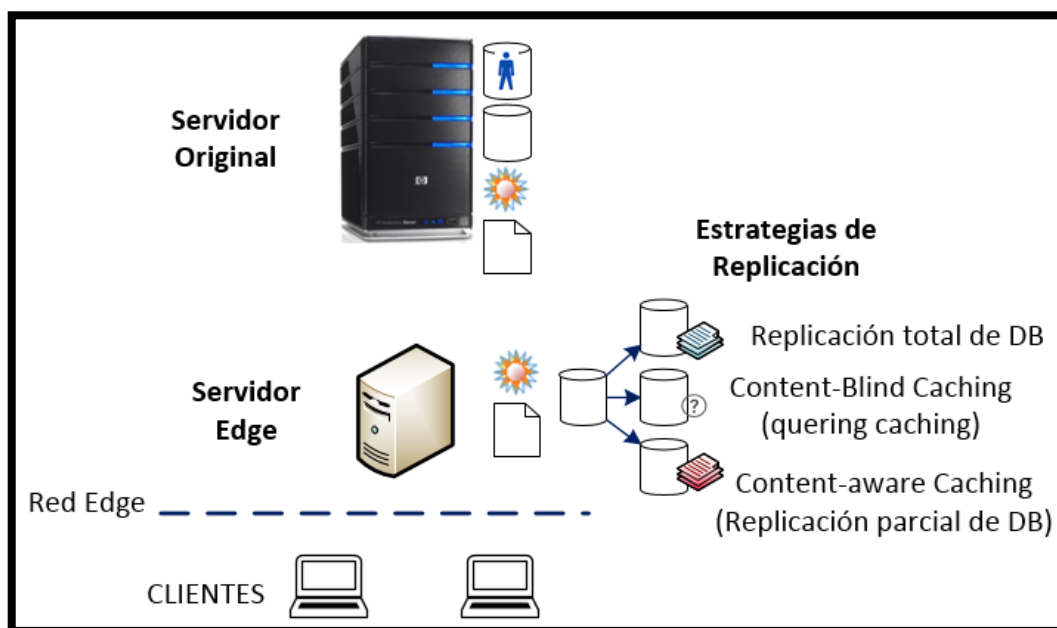


Figura: 4.22. La réplica de la capa de fondo en el servidor Edge [11]

- **Contenido Blind Caching:** Cuando se adopta el almacenamiento en caché de contenido *Blind*, los servidores de borde caché son resultados de las consultas previas a la base de datos. De tal manera, el servidor puede procesar localmente futuras consultas idénticas o similares, mejorando así el rendimiento y el descongestionando la carga en la capa de back-end en el servidor de origen.
- **Contenido Aware Caching:** En este caso cada servidor perimetral ejecuta su propio servidor de base de datos, que contiene una vista parcial de la base de datos centralizada. El enfoque típico para la replicación de datos parcial es para empujar la sección de la base de datos cerca del borde, de acuerdo con los patrones de acceso. Puesto que el objetivo es mejorar el tiempo de respuesta percibido por el usuario final, los algoritmos para la colocación de por lo general incluyen la latencia de red en el modelo de rendimiento.

- **Full Database Replication:** Replicación completa base de datos mantiene copias idénticas de la base de datos en múltiples ubicaciones. Al mover una copia de la base de datos a los servidores de borde y mantener las copias de bases de datos coordinados entre ellos, se hace posible la entrega de contenido dinámico completamente en el borde de la red, sin la necesidad de modificar cada aplicación desplegada. Sin embargo, la gestión de la replicación de bases de datos presenta problemas de gestión consistencia graves, que son especialmente críticos para resolver cuando el cliente solicita desencadenan cambios frecuentes en los datos persistentes.

La solución más sencilla para administrar la replicación de bases de datos en entornos Web se basa en una copia primaria centralizada en el servidor de origen y replicadas copias secundarias en los servidores de borde. Sólo lectura transacciones pueden ejecutarse completamente en el borde mediante el acceso a la copia de la base de datos secundaria local. Sin embargo, para las transacciones que requieren operaciones de actualización (como en la escritura en su mayoría escenarios), todos los accesos de bases de datos son redirigidos a la copia principal base de datos ubicada en el servidor de origen centralizado. La base de datos primaria se propaga cualquier actualización de las bases de datos secundarios sobre una base regular. Un primer inconveniente de este enfoque es que los servidores de borde deben ser conscientes de la semántica de la aplicación, ya que tiene que saber si una solicitud activa una actualización o una transacción de sólo lectura. “Además, en esta solución se mantiene la consistencia de los datos replicados a través de un esquema de propagación de actualización perezoso, que presenta dos efectos negativos. En primer lugar, los datos a los servidores de borde podrían ser obsoletos. En segundo lugar, un accidente podría causar una pérdida de datos” [11].

En un CDN las réplicas de bases de datos están geográficamente distribuidas en una red WAN. Si la aplicación web genera un número significativo

de cambios de base de datos, una gran cantidad de tráfico puede sobrecargar la red de área amplia y el impacto negativo en el rendimiento, ya que cada actualización necesita ser propagado a todas las demás réplicas para mantener la consistencia de los datos replicados.

4.5 Perfil de usuario capa de replicación

La capa de perfil de usuario se basa en una base de datos para el almacenamiento de datos, como la capa de fondo. Sin embargo, los patrones de acceso para esta capa son muy diferentes si se compara con la capa de fondo.

El usuario normalmente interactúa con sólo un servidor de borde, por lo tanto, el perfil de un usuario dado se accede por un servidor de borde para toda la duración de una sesión de usuario. Este patrón de acceso tiene un impacto significativo en las políticas de consistencia y replicación. De hecho, todo el conjunto de datos de perfiles de usuario se puede dividir y se distribuye a través de los nodos de borde en función de los patrones de acceso de usuario. Dado que no se necesita la replicación, problemas de coherencia se limitan a garantizar que los perfiles de usuario en los servidores de borde son consistentes con los datos en el servidor de origen.

Sin embargo, vale la pena tener en cuenta que, incluso si el usuario accede a un único servidor de borde para toda la duración de su período de sesiones, la migración de usuarios entre múltiples servidores de borde puede ocurrir entre sesión consecutiva. Por lo tanto, es necesario garantizar que la migración de datos de perfil de usuario después de que el usuario, como se muestra en la Figura: 4.23. El apoyo a este comportamiento no se optimiza de forma explícita en la mayoría de las estrategias de replicación de datos de back-end.

Actualmente, el perfil de usuario se actualiza o bien manualmente por el usuario a través de formularios basados en la web o se actualiza automáticamente por el sistema Web sobre la base del comportamiento del usuario. “La información almacenada en el perfil de usuario y la manera de recogerlos depende de los servicios basados en la Web que han de ser desplegado” [11].

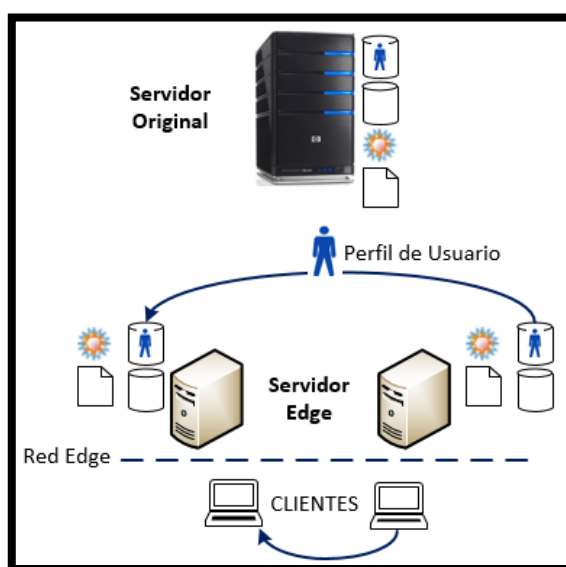


Figura: 4.23. La réplica de la capa de perfil de usuario en el servidor Edge [11]

- **Generación de contenido personalizado a través de la agregación de las fuentes de datos externas:** Este servicio es común en múltiples portales personalizados como myYahoo, iGoogle que proporcionan al usuario un servicio de noticias personalizado. El perfil de usuario contiene información sobre la que se alimenta de interés para los usuarios y sobre cómo el contenido personalizado se va a presentar.
- **Filtrado colaborativo.** Este tipo de servicio es compatible con la interacción de los usuarios que proporcionan información sobre otros

usuarios o temas. Este tipo de generación de contenido personalizado se utiliza a menudo en los sistemas de recomendación que proporcionan sugerencias sobre los bienes a adquirir, basado en el análisis de los comportamientos de usuario similares, o para clasificar la reputación de un usuario en una red social.

- ***Ubicación y servicios basados en los alrededores.*** Estos servicios generan contenido personalizado sobre la base de la ubicación geográfica del usuario. La posición del usuario se determina a través del análisis de información relacionada con los datos, o se suministra explícitamente por el usuario al acceder al servicio. La ubicación del usuario se compara con los datos geográficos y la generación y entrega de contenido estático y dinámico (por ejemplo, consultas) se lleva a cabo de acuerdo con la ubicación del usuario y sus alrededores, posiblemente combinado con las preferencias del usuario.

Estos ejemplos muestran que, incluso si cierta información puede ser proporcionada explícitamente de los usuarios, una fracción significativa de los datos almacenados en los perfiles de usuario se infiere a través de la minería de datos de archivos de registro, las cookies y el historial del usuario clic. Con las tecnologías de recopilación de información disponibles, es posible extraer información interesante en relación con los usuarios, incluidos los datos confidenciales, tales como rasgos políticos, físicos y sexuales. Por otra parte, la mayoría de las técnicas son casi transparente para los usuarios que a menudo son totalmente inconscientes. No autorizado de recogida de información a los usuarios se produjo en los últimos años, por ejemplo, por el servicio de publicidad comercial doubleclick.com. Varios servicios comerciales, incluyendo los motores de búsqueda, se asociaron con doubleclick.com. Los sitios comerciales utilizan cookies para supervisar sus actividades de los visitantes, y cualquier información recogida se almacena en bases de datos doubleclick.com. Estos perfiles de

usuario se utilizaron luego por doubleclick.com para seleccionar los banners de publicidad más adecuados para los usuarios.

4.6 Simulación de la arquitectura CDN

Al ser una tecnología que está actualmente en desarrollo, no hay muchos programas que nos ayuden a estructurar y diseñar en totalidad el comportamiento de una CDN, mediante la investigación efectuada se encontró el software CDNSim.

CDNSim fue desarrollado por el Departamento de Informática de la Universidad Aristóteles de Salónica en Grecia y ha sido diseñado para proporcionar una simulación realista de CDN, simulando los servidores sustitutos, el protocolo TCP / IP, y las funciones principales de CDN.

Para poder ejecutar el programa es necesario tener instalado el sistema operativo Ubuntu 9.10 sobre el cual se hicieron las respectivas pruebas con el programa en mención.

Para ejecutar CDNSim, abrimos un terminal y comenzamos con la instalación del programa:

- Ingresamos a la ubicación donde se encuentra la carpeta que contenga el programa (para este caso /home/diego/CDNSim) y digitamos los siguientes comandos para descomprimir el archivo:

```
tar xf CDNSim.tar
cd CDNSim
```

- Una vez que lo tenemos descomprimidos procedemos con la instalación del programa usando los siguientes comandos:

```
./ubuntu.sh
```

O también podemos instalar de forma separado todos los paquetes necesarios para que se pueda ejecutar correctamente CDNSim los cuales son:

- *g++*
- *bison*
- *boost Libraries and their development headers*
- *python wxgtk at Least 2.8*

- Al momento de la instalación debemos verificar que no haya existido ningún error y procedemos a ingresar la siguiente instrucción:

```
./runme.sh
```

Si no se presente ningún error durante el proceso de instalación ya estamos listos para proceder a ejecutar el programa y proceder con las pruebas.

Para ejecutar el programa ingresamos al terminal y ejecutar la siguiente sentencia:

```
cd /home/diego/CDNsim/CDNsimgui  
python CDNsim.py
```

La primera pantalla que aparece define la política de redirección CDN. El cliente es redirigido a un servidor alternativo de acuerdo con la política. Después de un error de caché el servidor sustituto es redirigido también de acuerdo con la misma política

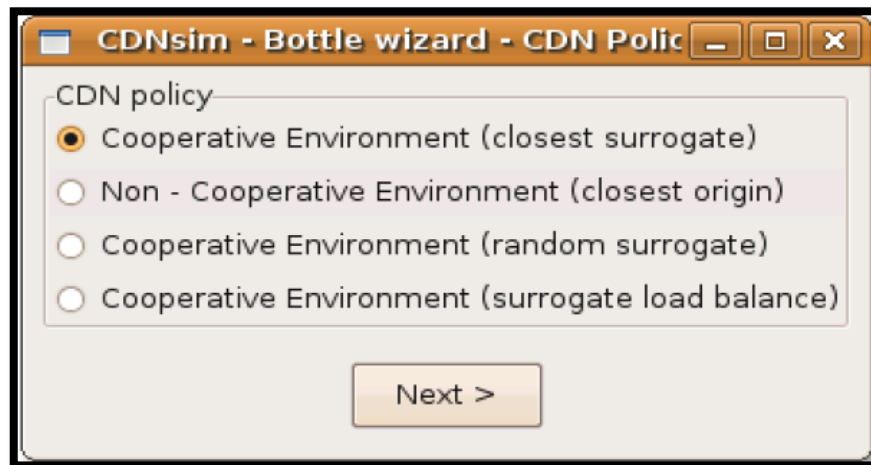


Figura: 4.24. En el CDNSim muestra política de redirección CDN

- **Cooperative Environment** (Cooperación ambiental).- “El cliente se redirige al servidor más cercano de sustitución en términos de saltos de red. Después de un error de caché, el servidor sustituto recupera el objeto del servidor más cercano de la madre sustituta alternativa que contiene el objeto solicitado. El objeto se almacena en la cache y luego se sirve al cliente” [12].
- **Non - Cooperative Environment** (No cooperativo del medio ambiente).- “El cliente se redirige al servidor más cercana de sustitución en términos de saltos de red. Después de un error de caché, el servidor sustituto recupera el objeto desde el servidor más cercano de origen, la almacena en la memoria caché y lo sirve al cliente” [12].
- **Cooperative Environment** (Cooperativo para el medio ambiente).- El cliente es redirigido a un servidor sustituto al azar. Después de un error de caché, el servidor sustituto recupera el objeto de un servidor al azar sustituto alternativo que contiene al objeto solicitado. Si el objeto no es compartido en todos los servidores sustitutos, entonces el servidor sustituto recupera el objeto de un servidor de origen aleatorio.

- **Cooperative Environment** (Cooperativo para el medio ambiente).- El cliente se redirige al servidor más cercano sustituto en términos de saltos de red. Si el servidor sustituto se carga a 95%, entonces el cliente se redirige al servidor sustituto menos cargado. Después de un error de caché, el servidor sustituto recupera el objeto desde el servidor más cercano que contiene el objeto solicitado. Una vez más, si la carga es de ~ 95% el servidor sustituto es remitido a un servidor de alquiler menos cargado que contiene el objeto. El objeto se almacena en la caché y luego se sirve al cliente.

La siguiente pantalla proporciona opciones para la topología de la red. CDNSim puede construir cualquier tipo de topología de red cableada, siempre y cuando se proporciona la configuración apropiada. Configuraciones avanzada se puede realizar mediante la edición del archivo *base.ned*. En este proyecto tratamos de simplificar el procedimiento, ofreciendo algunas opciones básicas:

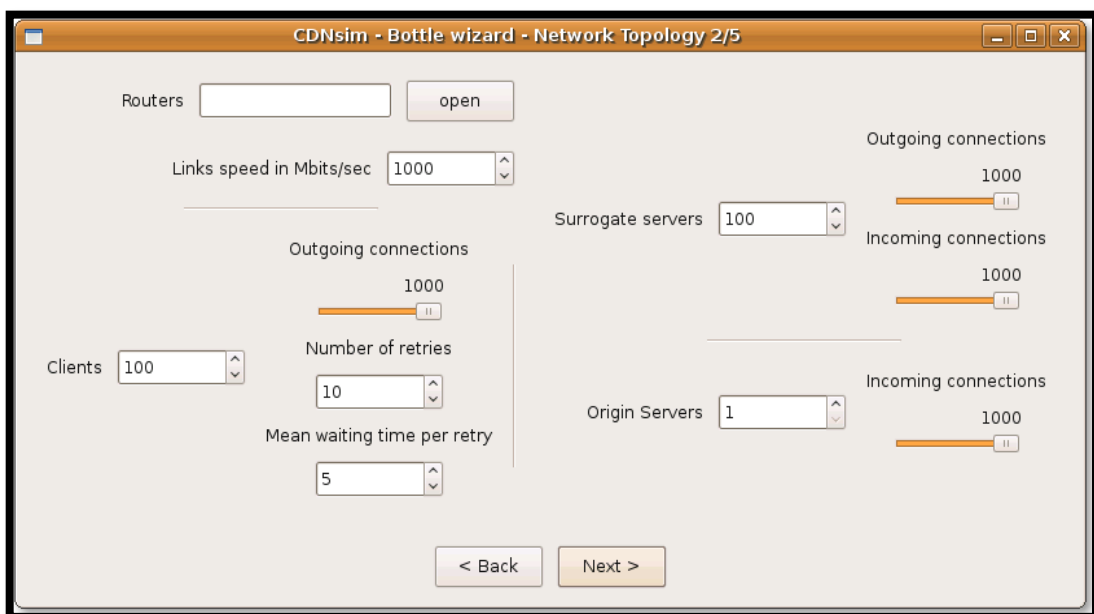


Figura: 4.25. Opciones para la topología de la red en CDNSim

- **Routers.** El usuario debe proporcionar un archivo de texto que describe la topología de la red troncal de los routers. El formato del archivo es "int int \ n" sin las comillas. Cada entero representa un identificador de nodo, cada línea representa un enlace entre los dos nodos y "\ n" es el carácter de nueva línea. No hay necesidad de definir enlaces "dobles", es decir, 1 2 y 2 1. El comportamiento por default es crear enlaces dobles. Los ID de nodos deben estar en el rango [0, number_of_routers). Todos los otros nodos de red (servidores sustitutos, los servidores de origen, clientes) están unidos aleatoriamente a la columna vertebral routers. Si usted proporciona las mismas opciones usted conseguirá la misma topología de la red cada vez que se ejecuta el asistente. La Tabla: 4. 1. muestra un ejemplo cómo se forma el archivo.

Tabla: 4. 1. Ejemplo de Archivo Router (random50)

Router	Nodo
0	46
1	8
2	44
2	38
2	25
3	34
3	19
3	10

- **Velocidad de enlace.** Esta opción define la velocidad de cada enlace. Todos los enlaces tendrán la misma velocidad. La creación de una red con distintas velocidades requieren una edición manual del archivo generador de topología de red.
- **Número de clientes.** Este es el número de clientes que existen en la red.

- **Número de Clientes de conexiones salientes.** Cada cliente puede manejar más de una conexión de solicitud de objetos. Esta opción establece un límite superior.
- **Número de Clientes de reintentos.** Cada cliente puede volver a intentar que muchas veces en caso de solicitud falló.
- **Media Clientes vez por reintento de espera.** A raíz de la distribución exponencial con esto decir la cada cliente permanece inactivo hasta el próximo reintento.
- **Número de servidores sustitutos.** Este es el número de servidores sustitutos que tiene la CDN.
- **Número de servidores sustitutos de conexiones salientes.** Si el caché pierde los servidores sustitutos tienen que recuperar los objetos solicitados desde los servidores alternativos. Esta opción define el número máximo de tales conexiones. La denegación de servicio se produce si se cumple este límite.
- **Número de servidores sustitutos de conexiones entrantes.** Esto establece el número máximo de conexiones de un sustituto puede recibir de otros servidores o clientes sustitutos para los objetos. La denegación de servicio se produce cuando se alcanza el límite.
- **Número de servidores de origen.** Define el número de servidores de origen en la red. Los servidores de origen contienen todo el sitio web y que son el último recurso para una solicitud que deberá satisfacerse.
- **Número de conexiones entrantes servidores de origen.** Define el número de peticiones de entrada para los objetos de los servidores de

origen que se pueden manejar simultáneamente. La denegación de servicio se produce si se supera este límite.

En la siguiente pantalla el usuario introduce el sitio web y el tráfico de archivos en los clientes. Más específicamente:



Figura: 4.26. Sitio web y el tráfico generado por el usuario

- **Sitio Web.** Este archivo contiene todos los objetos existentes en el sistema y se puede transferir. El formato es "int int \ n", donde el primer número entero es un identificador de objeto único y el segundo entero es el tamaño en bytes. Los id de objeto deben estar en el rango [0, number_of_objects). La Tabla: 4.2. muestra un ejemplo cómo se forma el archivo.

Tabla: 4. 2. Ejemplo de archivo Web (web_site16000)

Objetos	Bytes
0	388637
1	71125
2	58512
3	101924
4	106333
5	368734
6	299117
7	108522

- **Traffic.** Este archivo describe cual los clientes, cuándo y cuáles harán una solicitud a la CDN. El formato del archivo es "doble int int \ n". El primer número es la fecha y hora de la solicitud, el segundo es el cliente y el último es el objeto solicitado. La marca de tiempo toma valores ≥ 0 y el archivo debe ser ordenado por el aumento de fecha y hora. Los clientes toman valores ≥ 0 . Si el número de clientes están en el rango $[0, \text{number_of_clients})$ donde `number_of_clients` es el definido anteriormente, entonces le damos Aceptar. De lo contrario, los números de cliente en el archivo de rastreo se vuelven a asignar al azar para cumplir con el rango $[0, \text{number_of_clients})$ pero conservando las peticiones de cada cliente se le asigna inicialmente a. Si usted proporciona la misma entrada obtendrá la misma asignación de cada vez que se ejecuta el asistente. Los objetos solicitados deben estar en el rango $[0, \text{number_of_objects})$. La Tabla: 4.3. muestra un ejemplo cómo se forma el archivo.

Tabla: 4. 3. Ejemplo de archivos Tráfico (trace_file50000)

Tiempo y Fecha	Clientes	Objeto
2	25097	10250
9	58093	11159
15	21390	3229
15	65730	1157
21	77012	269
23	21390	3110
28	58093	11238
28	77854	10350

Existe una utilidad que convierte un registro de Apache en un sitio web CDNsimsim y un archivo de tráfico CDNsimsim. Esto se puede encontrar en CDNsimsim / CDNsimsim / Apache2CDNsimsim. Es la NASA2CDNsimsim.py. Está diseñado para analizar los registros del servidor web de la NASA, esta utilidad está lejos de ser perfecta ya que necesita de una afinación avanzada. De todos modos, el comando para

generar el archivo es `./NASA2CDNsim.py -i APACHE_LOG - CDNsimtrace -s CDNsimWebsite`

En este paso configuramos cada servidor sustituto. La configuración es establecida por un archivo que describe el contenido, la capacidad y la política de reemplazo de caché de cada servidor sustituto.



Figura: 4.27. Configuración del comportamiento de los servidores sustitutos

El archivo contiene registros, cada uno referido a un servidor de alquiler. Un ejemplo de este tipo de registro es:

```
0
1
0
LRU
856233632.6
```

- **0.-** Los registros en el archivo, deben seleccionarse mediante el aumento de índice de servidores sustitutos.
- **1.-** se lo debe colocar siempre 1.
- **0.-** El usuario puede elegir un nombre de archivo distinto de 0. Si es NULL, entonces indica que el caché se vacía al principio y por lo tanto no

hay archivo proporcionado. El archivo contiene las líneas en la forma "char int \ n" o "int \ n". El número entero es el identificador de objeto que se almacena en el comienzo de la simulación y se extendieron en [0, number_of_objects).

- El char puede obtener el valor de "s" o "v". "S" significa que el objeto se almacenarán al principio y no se eliminará hasta el final de la simulación. "V" significa que el objeto se almacena, pero puede ser removido en cualquier momento si la política lo permite.
- **LRU.-** La política de reemplazo LRU caché. Por el momento sólo está disponible LRU.
- **856233632.6.-** Muestra el tamaño de la caché en Bytes.

Aquí el usuario debe proporcionar la entrada para el siguiente:

- **Directorio de salida.** Este es el directorio donde se almacenará la salida (no de los resultados de la simulación).
- **Nuevo nombre de botella.** Nombre descriptivo para el nuevo experimento.

En este paso toda la entrada proporcionada en los pasos anteriores y todas las librerías necesarias son verificadas y empaquetadas dentro de un archivo comprimido llamado botella. Si hay algo que no es válido, entonces obtendrá un mensaje de error. Usted puede abrir el archivo generado (es formato tar.gz). No cambie nada menos que sepa lo que está haciendo. Se puede realizar modificaciones avanzadas de la topología de red mediante la edición del archivo *base.ned* dentro de las botellas.



Figura: 4.28. Creación del Bottle y donde se almacenará

Después de haber terminado de crear el Bottle es necesario cerrar la aplicación para poder seguir trabajando en la terminal.

Para poder obtener un reporte de cada uno de los *Bottle* es necesario seguir una serie de pasos, los cuales nos arrojará un archivo HTML donde nos indica una serie de datos importantes.

Se tienes que copiar del directorio *CDNSim/CDNSim/CDNSimbatch* el archivo *batch.sh* al directorio donde se almaceno el *Bottle* y después se debe ejecutar el siguiente comando:

```
./batch.sh ./
```

Este comando tardara algunos minutos dependiendo de la topología creada, como resultado arrojará un archivo con terminación **.TAR** y con el mismo nombre.

Para terminar, el directorio *CDNSim/CDNSim/CDNSimstats* contiene tres archivos, *stats.sh*, *stats.py* y *util.py*.

stats.sh utiliza el *stats.py* y *util.py* para elaborar las estadísticas. El concepto es generar un informe HTML que resume los resultados de cada botella, para eso se debe de copiar los tres archivos en donde se tiene el Bottle (donde se copió el

archivo Batch.sh) y después ejecutar el siguiente comando estando en el directorio donde se pusieron las archivos (/home/diego/Bottle/salida):

```
./stast.sh /home/diego/CDNsim/Resultados  
/home/diego/CDNsim/Resultados /stast.py  
/home/diego/CDNsim/Resultados util.py
```

Con este último paso podemos visualizar los resultados enviados por el simulador por medio de una página HTML.

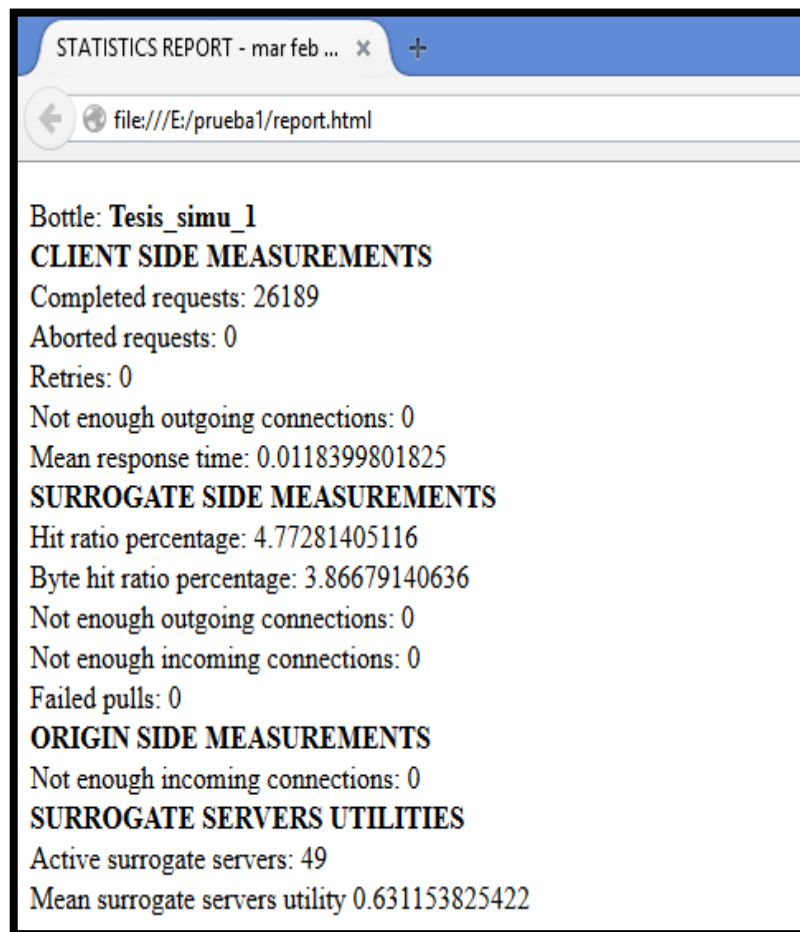


Figura: 4.29. Resultados desplegados vía web

4.7 Análisis de resultados de la simulación

En este proyecto se probó la funcionalidad del software CDNSim para el cual tomamos los siguientes archivos random_50, web_site16000, tracefile_1500 y pushBased_LRU_STATIC_100ss con los cuales se efectuaron las simulaciones, dichos archivos están disponible en la página del autor y los cuales recomiendan usar.

La simulación se enfocó en varia la cantidad de usuarios con el fin de verificar si varia el tiempo de carga y determinar si en algún momento se los clientes alcanzarían al servidor original.

En la Figura: 4.30. se muestra un esquema de ejemplo general de cómo CDNSim está formando las conexiones entre los routers, los servidores y los clientes con el fin de emular una red CDN real.

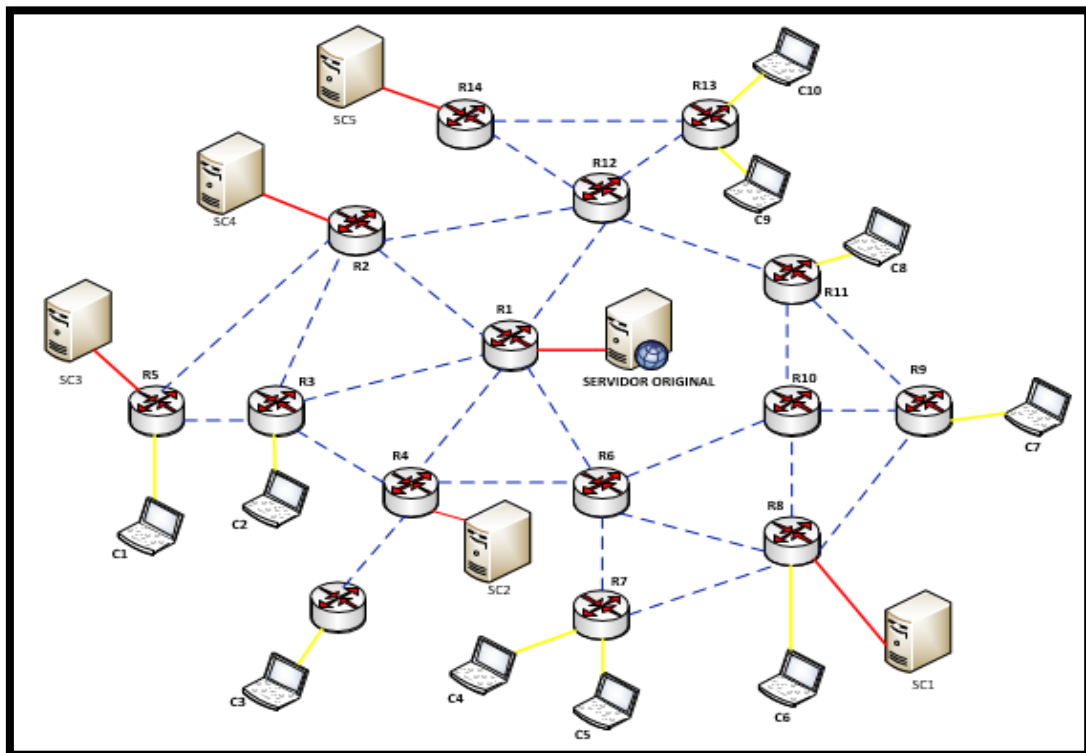


Figura: 4.30. Esquema Ejemplo de conexión simulador CDN

Los resultados obtenidos con el simulador en el lado del cliente fue que se comprobó que pese a la variación de densidad de clientes, el tiempo de respuesta se ha mantenido y sus variaciones han sido en milésimas como se muestra en la Tabla: 4.4., lo cual muestra la efectividad de uso de recurso de CDN, debido a que trata de siempre buscar el servidor replica más cercano al cliente y nunca vemos saturado ningún servidor replica ya que se distribuyen los clientes alrededor de toda la red.

Tabla: 4.4. Resultados vistos del lado del cliente

CLIENTE	
Usuarios	Tiempo de Respuesta
100	0,0129
200	0,0134
300	0,0136
400	0,0147
500	0,0151
600	0,0139
700	0,0139
800	0,0147
900	0,0144
1000	0,0140

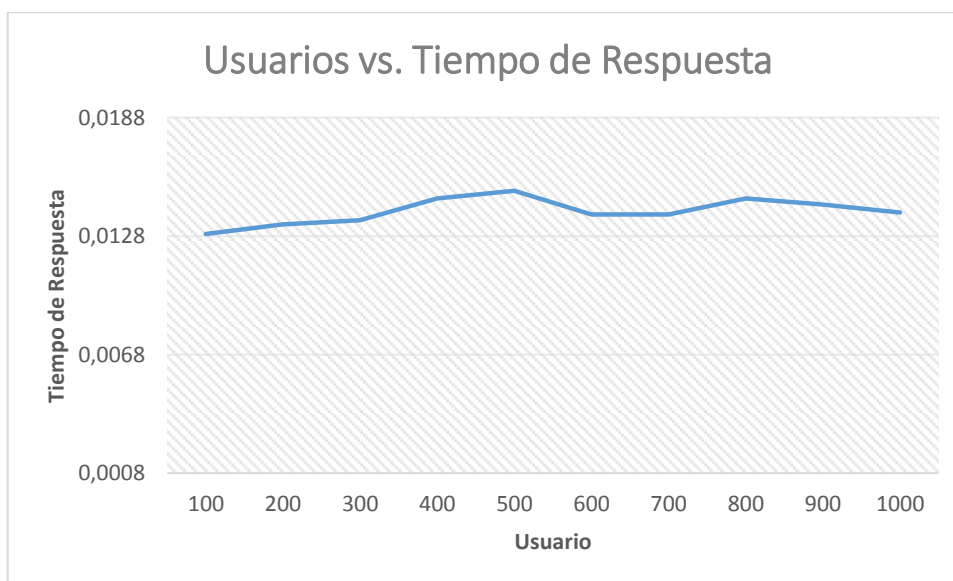


Figura: 4.31. Resultados de Tiempo de respuesta al aumentar la densidad de usuarios

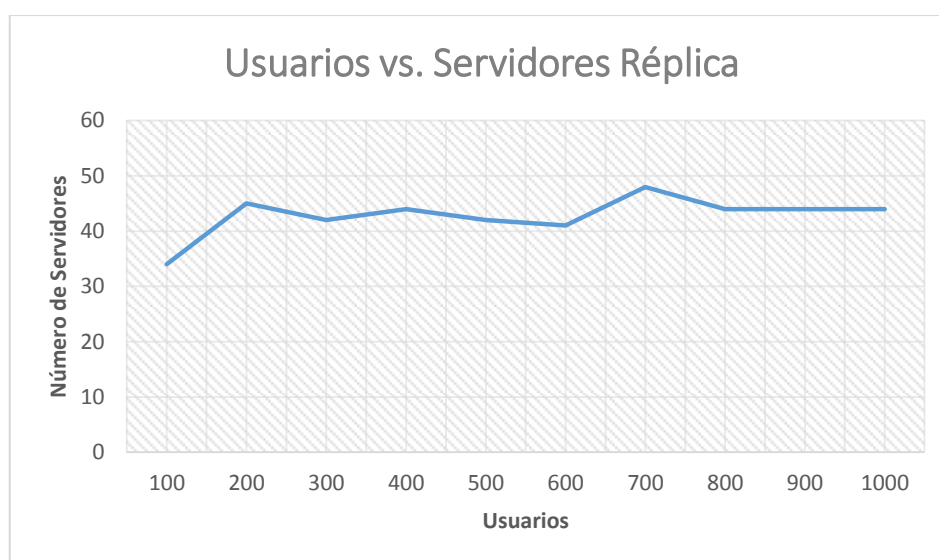
En la Figura: 4.31. podemos ver un promedio donde demostramos que la cantidad de usuarios en nuestra red no afecta en el tiempo de respuesta hacia los clientes ya que mantiene en un promedio de alrededor de 0,01230 lo que demuestra la efectividad de la CDN.

Además otro dato importante que podemos determinar a partir de esta simulación es la cantidad de servidores replicas usados para satisfacer cierta cantidad de usuarios nunca llegan a superar el 50%, es decir que para un promedio de 100 hasta 1000 clientes solo se llegan a utilizar 44 servidores para satisfacer todas las peticiones simultáneas realizadas por los clientes.

En la Tabla: 4.5. podemos apreciar los resultados vistos del lado del servidor replica y observamos que el tiempo de respuesta hacia los clientes no varía mucho, sin importar la cantidad de usuarios que se tengan, siempre se prioriza la distancia más corta hacia el servidor replica más cercana al cliente.

Tabla: 4. 5. Resultado del uso de servidores réplicas dependiendo del número de usuarios

SERVIDORES RÉPLICAS		
Usuarios	Servidores Usados	Período de Uso
100	34	0,6426
200	45	0,5881
300	42	0,5711
400	44	0,5685
500	42	0,5416
600	41	0,5673
700	48	0,5487
800	44	0,5395
900	44	0,5446
1000	44	0,5519

**Figura: 4.32.** Gráfica que demuestra la cantidad de servidores que se debe usar para un determinado grupo de usuarios

En la Figura: 4.32. podemos analizar que el uso de los servidores replica nunca van a superar para este caso el 50 % por más cantidad de usuarios que tengamos presentes en nuestra red e incluso nunca se vio la necesidad de llegar con una solicitud al servidor central.

Los resultados totales obtenidos en las simulaciones se las adjuntan en el Anexo 1.

La conclusión que podemos determinar sobre este simulador es que nos demuestra la funcionalidad y utilidad de CDN en una red de estilo malla. Además de ser una aplicación que nos ayuda a dimensionar la cantidad de servidores replicas a ser usados, dependiendo de la cantidad de usuarios y de routers que tengamos en nuestra topología.

Con esto se verifico ciertas cualidades que CDN plantea como beneficios de su tecnología:

- La entrega de contenido es más veloz.
- Ahorro de tiempo a clientes finales.
- Evita tantos saltos en la red, encuentra a la réplica más cercana.
- Elimina el congestionamiento de información sobre la red.

4.8 Equipamiento CDN

Mediante la investigación que se ha realizado en torno a este tema, se ha logrado determinar cierto equipamiento el cual podría ser utilizado para el diseño e implementación de una arquitectura CDN.

4.8.1 Decoder ATEME DR8400

DR8400 es un Universal Integrated receptor / decodificador, este equipo apoya todo el perfil desde High422P incluyendo MPEG-4 4: 2: 2 de 10 bits.

DR8400 está diseñado para los mercados donde la calidad y rendimiento de primera calidad son más importantes que nunca.

El DR8400 por defecto tiene 3 tipos de entrada diferentes: built-in DVB-S / S2 demodulador, IP Gigabit y ASI con una velocidad de bits TS de hasta 200Mbps. Combinando la velocidad de bits de decodificación podemos alcanzar 80Mbps sin comprometer la CABAC, el DR8400 está abordando un amplio espectro de aplicaciones principalmente en el campo de las CDNs.



Figura: 4.33. DR8400

Características y ventajas principales:

- SD only or SD/HD H.264/ MPEG-4 AVC, 4:2:0 / 4:2:2 8-bit /10-bit support
- MPEG-2 4:2:0 / 4:2:2
- Two Built in RF demodulators with fast-frequency locking
- 8 audio stereo pairs decoded
- Pro MPEG FEC and BISS – 0/1/E support
- IP output Stream forwarding
- HD to SD Broadcast quality downscale
- Simultaneous SD and HD output
- Thumbnail and Advanced Status monitoring
- Full VBI support
- SNMP support

- H.264 and DVB-S2 advantage with up to 50% bandwidth saving on contribution links
- With legacy MPEG-2 support, the DR8400 gives great flexibility on the field
- DR8400 is upgradeable to 1080p60, providing the best CAPEX reduction

4.8.2 Encoder Cisco DCM 9900

Sistemas digitales de hoy exigen soluciones potentes, flexibles y compactas que permiten al proveedor de servicios de apoyo a las nuevas arquitecturas de red. El DCM D9900 es un Gestor de Contenidos Digitales Procesados (DCM), es una plataforma 2RU compacto capaz de procesamiento de un gran número de secuencias de vídeo MPEG. La Serie D9900 es la nueva generación de equipos de procesamiento de cabecera inteligente donde la combinación de capacidad y flexibilidad conduce a una solución rentable.



Figura: 4.34. Cisco DCM 9900

Características y Ventajas:

- Dense MPEG-2 to H.264 transcoding for IPTV HE and cable video over DOCSIS® applications

- Can be housed in existing DCM chassis and coexists with existing DCM cards (requires Compact Flash upgrade to 16 GB)
- Up to 48 SD / 12 HD channels in 2RU chassis
- Process both SD and HD in the same chassis
- Can be combined with other DCM functionality, e.g., DVB scrambling, BISS-1 (de)scrambling, FEC

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- La investigación realizada fue un intento de abordar algunos de los principales aspectos de las arquitecturas existentes para la CDN. Con el constante desarrollo de la Internet se ha visto la necesidad de encontrar nuevos métodos para descongestionar la red de ahí que analizamos los beneficios que CDN nos puede brindar como servidor de contenidos, en este proyecto se estudió la arquitectura y su funcionamiento con el fin de comenzar a adentrarnos un poco en esta nueva arquitectura en el entorno de la red.
- Se Analizó y categorizó la CDN según la funcional y los atributos no funcionales. Los puntos que tratados fueron composición, distribución de contenido y gestión, y la medición del desempeño de una CDN. Además, identificamos tres cuestiones, a saber, la tolerancia a fallos, la seguridad, y la capacidad para la aplicación web de alojamiento como para introducir desafíos en el desarrollo de CDN.

- La industria de la CDN se está consolidando como un resultado de las adquisiciones o fusiones de metodologías y algoritmos. En consecuencia, las técnicas de distribución de contenidos, almacenamiento en caché y replicación están ganando más atención con el fin de cumplir con los nuevos requisitos técnicos y de infraestructura que demanda el constante desarrollo de Internet.
- Las Redes de Contenido se encuentra aún en una etapa inicial de desarrollo en Ecuador, actualmente Level 3 ha sido quien ha dado el primer paso y ya se encuentra prestando servicios de CDN, además aún existe relativamente pocos trabajos que busquen formalizar un marcos conceptuales para el estudio de la arquitectura y funcionamiento. La lista de características relevantes presentadas en este trabajo puede ser entonces de ayuda a la hora de buscar entender el funcionamiento de la arquitectura y los requisitos necesarios para implementar este tipo de redes.
- CDN se basan en una distribución proactiva de las réplicas de contenido a los servidores distribuidos geográficamente cerca del borde de la red, en la proximidad de los usuarios finales. CDN por lo tanto son sistemas complejos. Su comprensión y diseño implica el conocimiento en muchos campos de investigación: la medición del tráfico de Internet, caché de contenido, solicitud de cambio de ruta en los distintos niveles de protocolos de comunicación, control de admisión de la solicitud, balanceo de carga y más, dependiendo del tipo de contenido y la aplicación considerada.
- Se analizó y estudio la categorización de los atributos técnicos y de infraestructura de redes de entrega de contenido (CDN) basándonos en cuatro aspectos: composición CDN, de distribución de contenidos y de gestión, de peticiones y de enrutamiento, y la medición del desempeño. Con el fin de obtener una visión de la tecnología, los

servicios, las estrategias y las prácticas que se siguen actualmente en este campo.

5.2 Recomendaciones

- El desarrollo de la CDN es una solución inminente ante el crecimiento del internet por eso resulta de mucha importancia que se ahonde más en el tema y se busquen relatar pruebas prácticas para verificar su funcionalidad con el objetivo de una posible implementación.
- Se recomienda a los encargados del área de las TIC en la Universidad de las Fuerzas Armadas ESPE que se aproveche el enlace que se tienen con Level 3 y se solicite una CDN de prueba para apreciar los beneficios de la misma.
- La Universidad de las Fuerzas Armadas ESPE al trabajar con contenido web demasiado pesado como es el generado por la televisión digital la aplicación de una CDN podría acelerar los tiempos de entrega del contenido, mejorar sus rankings de búsqueda y aumentar la fidelidad de los clientes.
- La simulación realizada con el software CDNSim es el primer paso en el estudio de esta nueva arquitectura, se recomienda probar y experimentar nuevas cosas con dicho programa ya que al ser un software creado en Linux nos da una gran apertura para modificar archivos y cambiar parámetros de la simulación.
- Se recomienda visitar el siguiente sitio web [Google Page Speed](#) para verificar si la web se carga en un tiempo aceptable y, así plantearnos si es necesario usar una arquitectura CDN para mejorar el rendimiento de nuestra Web.

BIBLIOGRAFÍA

- [1] N. Bartolini, E. Casalicchio y S. Tucci, *A Walk through Content Delivery Networks*, Roma: Universit`a di Roma, 2004.
- [2] R. Buyya y M. Pathan, *Content Delivery Networks*, Australia: Athena Vakali, 2008.
- [3] P. R. Bocca y H. C. Bosi, *Redes de contenido: un panorama de sus características y principales aplicaciones*, Motevideo, 2003.
- [4] G. Peng, *CDN: Content Distribution Network*, New York: Stony Brook, 2008.
- [5] Microsoft, *Developer Network*, 2004. [En línea]. Available: <http://msdn.microsoft.com/en-us/library/ms812590.aspx>.
- [6] Network Working Group , *Protocolo de Caché de Internet (ICP)*, versión 2, Septiembre 1997. [En línea]. Available: <http://www.rfc-es.org/rfc/rfc2186-es.txt>.
- [7] G. Pallis, A. Vakali, K. Stamos, A. Sidiropoulos y D. Katsaros, *A Latency-based Object Placement Approach in Content Distribution*, *IEEE*, p. 8, 2005.
- [8] Y. Chen, *Dynamic, Scalable, and Efficient Content*, de *Content Delivery Networks*, Greece, Springer , 2008, p. 25.
- [9] *Programación en Castellano*, 08 12 2014. [En línea]. Available: http://programacion.net/articulo/desarrollo_de_aplicaciones_web_con_tapestry_334.
- [10] TechTarget, *Multicast Dissemination Protocol (MDP)*, 08 12 2014. [En línea]. Available: <http://searchnetworking.techtarget.com/definition/Multicast-Dissemination-Protocol>.
- [11] C. Canali, V. Cardellini, M. Colajanni y R. Lancellotti, *Content Delivery and Management*, de *Content Delivery Networks*, Grecia, Springer, 2008, p. 25.
- [12] *CDNsim tutorial*, *CDNsim tutorial*, 4 Enero 2015. [En línea]. Available: <http://oswinds.csd.auth.gr/cdnsim/docs/tut1/index.html>.

ACTA DE ENTREGA

El presente proyecto de grado fue entregado al Departamento de Eléctrica y Electrónica y reposa en la Universidad de las Fuerzas Armadas – ESPE desde:

Sangolquí, 26 de Mayo de 2015

ELABORADO POR:



Diego Gerardo Zamora Cruz

Ci: 1718599432

AUTORIDAD:



Ing. Paúl Bernal

DIRECTOR DE LA CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES