



FACULTAD DE INGENIERÍA
INGENIERÍA CIVIL INFORMÁTICA

Una arquitectura caché para redes ICN basada en comportamiento de usuario

Tesis para optar al grado de ingeniero civil en informática.

Autor:

Mathias Nicolas Velilla Brandau.

Profesores guía:

Carlos Gomez-Pantoja
Miguel Gutierrez

Santiago de Chile, Chile.

Abril, 2017

Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Desafíos	3
1.3. Contribución de la tesis	3
1.4. Estructura de la tesis	4
2. Descripción del problema	5
2.1. Contextualización	5
2.2. El problema	5
2.2.1. Declaración del problema	5
2.2.2. Diagrama de ishikawa	5
2.3. Objetivos	5
2.3.1. Objetivo general	5
2.3.2. Objetivos específicos	5
2.3.3. Alcance de los objetivos	6
3. Marco teórico	7
3.1. Information Centric Network	7
3.2. Caché	11
3.2.1. Organización del Cache	12
3.2.2. Estructura del caché	13
3.3. Comportamiento de usuario	14
3.3.1. Ley de Zipf	14
3.3.2. Peticiones de usuarios	15
4. Estado del arte	19
4.1. NDNSIM	19
4.1.1. On Performance of Cache Policies in Named Data Networking [20]	19

4.2. Arquitectura de Caches	22
Bibliografia	22

Índice de figuras

3.1. Cronología de los principales hitos de ICN [25].	7
3.2. Arquitecturas de reloj de arena de Internet y NDN [3].	8
3.3. Paquetes NDN	9
3.4. Nodo CCN	10
3.5. Frecuencia de consulta periódica: "18 de septiembre (2004-2016)". Fuente: Google Trends.	16
3.6. Frecuencia de consulta ráfaga: "Muerte de Fidel Castro" (2004-2017). Fuente: Google Trends.	16
3.7. Frecuencia de consulta ráfaga: "Historia de Fidel Castro" (2012-2017). Fuente: Google Trends.	17
3.8. Frecuencia de consulta permanente: "YouTube" (2004-2017). Fuente: Google Trends.	17
4.1. Tabla de popularidad de contenido. Fuente: On Performance of Cache Policies in Named Data Networking [20]	20
4.2. Sección de la topología de la red. Fuente: On Performance of Cache Policies in Named Data Networking [20]	22
4.3. Proporción de hit en caché VS Tamaño caché. Fuente: On Performance of Cache Policies in Named Data Networking [20]	22
4.4. Rendimiento medio de la red VS Tamaño caché. Fuente: On Performance of Cache Policies in Named Data Networking [20]	23
4.5. Carga media del servidor VS Capacidad caché. Fuente: On Performance of Cache Policies in Named Data Networking [20]	23

Capítulo 1

Introducción

1.1. Motivación

El nacimiento del Internet en el año 1958 en EE.UU. a través de ARPA (*Advanced Research Projects Agency*) tuvo como objetivo la comunicación entre diversas entidades con propósitos investigativos. La aparición de la conmutación por paquetes implementadas en los mecanismos de comunicación sentó las bases para el desarrollo de esta nueva tecnología, la cual hoy en día ha revolucionado la informática y las comunicaciones como ninguna otra cosa convirtiéndose en una herramienta de índole mundial. Un mecanismo el cual nos permite diseminar información de manera inmediata, generando un medio de colaboración e interacción entre las personas y los ordenadores, desconociendo su ubicación física.

Durante los años 2000 - 2008 el uso diario del Internet en las personas americanas rondaba entre un 60 % y 70 % [Digital CITIZE]. Este crecimiento en el uso de la Internet, conlleva también a un aumento de la información presente en la web, dando como resultado el desarrollo de aplicaciones las cuales deben interactuar con un gran número de usuarios y analizando una sobresaliente cantidad de información. Motivo por el cual Internet impulsado por las demandas de las aplicaciones cada vez más emergentes y las capacidades de las nuevas redes de comunicación, se ha convertido en un mosaico arquitectónico que resulta en una creciente complejidad y vulnerabilidades imprescindibles, entregando como resultado violaciones de capas, proliferación de subcapas, y la erosión del modelo de extremo a extremo. Es bajo esta eventualidad que todos los cambios efectuados concluyen en un aumento de la complejidad, lo cual se traduce en una Internet osificada. Es por esta razón que los problemas anteriormente señalados no se encuentran directamente relacionados con los protocolos o mecanismo específicos del Internet actual, mas bien son causados esencialmente por

la incapacidad de integrar nuevos mecanismos. Esto quiere decir que los problemas son causados por la arquitectura de Internet y podrían ser resueltos con un nuevo diseño de arquitectura de Internet [18].

La creciente demanda de una distribución altamente escalable y eficiente del contenido ha motivado el desarrollo de futuras arquitecturas de Internet basadas en objetos de datos nombrados (NDO, por sus siglas en inglés), por ejemplo, páginas web, videos, documentos u otras informaciones. El enfoque de estas arquitecturas se conoce comúnmente como redes centradas en información (ICN). Por el contrario, las redes actuales se centran en el host, donde la comunicación se basa en hosts nombrados, por ejemplo, servidores web, PC, portátiles, teléfonos móviles y otros dispositivos[5].

Las arquitecturas ICN aprovechan el almacenamiento en red para el almacenamiento en caché, la comunicación multilateral a través de la replicación y los modelos de interacción que desacoplan a los remitentes y receptores. El objetivo común es lograr una distribución eficiente y confiable de los contenidos en donde cada uno de estos es nombrado única e independientemente desde la ubicación del productor(servidor), facilitando el almacenamiento en caché en los nodos intermediarios. Así, por ejemplo, los consumidores solicitaran contenidos enviando el denominado paquete de interés, el cual lleva el nombre del contenido, mientras que el productor o cualquier nodo que mantenga una copia del contenido puede responder a la petición realizada.[5]

A su vez, los usuarios ya antes mencionados como consumidores, realizan peticiones en la web, las cuales siguen una conducta dinámica caracterizándose por un elevado sesgo entre los diferentes conjuntos de peticiones, en otras palabras, dentro del universo de peticiones generadas existen conjuntos de peticiones que son regularmente solicitadas por los consumidores en intervalos de tiempos distintos, por lo contrario, otras peticiones escasamente son solicitadas. Dicho lo anterior, también existen situaciones dentro de un intervalo acotado de tiempo, donde surgen explosivamente peticiones las que se caracterizan por poseer un contenido en común, las cuales nacen por el desarrollo de un evento de interés popular teniendo como resultado un aumento sustancial en la demanda generada a los nodos, teniendo como consecuencia latencia y congestión en las redes.

1.2. Desafíos

Los desafíos que se afrontaran para la realización de proyecto de titulo I, son inicialmente el diseño de una nueva arquitectura de memoria caché para los nodos de las redes *ICN* (*Information Centric Network*), considerando el comportamiento de los usuarios por medio del tráfico de red. En segundo lugar, se debe incorporar una estrategia de caché(políticas de admisión, desalojo y reemplazo) y que será implementada en un simulador denominado *ndnSIM* (*Named Data Networking Simulator*), todo esto con el fin de mejorar resultados(Queresultados) en comparación con otras arquitecturas caché que se encuentran por defecto dentro del simulador(LFU, FIFO).

1.3. Contribución de la tesis

El aporte entregado por el proyecto de titulo, se puede identificar inicialmente por la creación diseño de una nueva arquitectura de memoria caché para los nodos de las redes *ICN* (*Information Centric Network*), considerando el comportamiento de los usuarios por medio del tráfico de red. En segundo lugar, se debe incorporar una estrategia de caché(políticas de admisión, desalojo y reemplazo) y que será implementada en un simulador denominado *ndnSIM* (*Named Data Networking Simulator*), todo esto con el fin de mejorar resultados de eficiencia de la memoria caché (Hit's, Miss) en comparación con otras arquitecturas caché que se encuentran por defecto dentro del simulador(LFU, FIFO). No obstante, a continuación se detallaran las contribuciones efectuadas por este trabajo:

- Diseño de una nueva arquitectura caché bajo el paradigma de las redes ICN, que contenga una subdivisión de tres grupos capaces de retener diferentes tipos de peticiones(intereses) de modo que la eficiencia del nodo se vea afectada positivamente. En cuanto a la división de la memoria caché, el primer segmento se encarga del almacenamiento de peticiones del tipo ráfaga, la segunda de guardar las de tipo permanente y para la ultima sección, la recaudación de peticiones de tipo variables.
- Creación de una topologia de redes ICN, utilizando el simulador ndnSIM, la cual contenga dentro de sus nodos la arquitectura caché anteriormente señalada

con la finalidad de inyectar tráfico de datos en base al comportamiento usuario para la obtención de resultados empíricos respecto a la nueva arquitectura caché diseñada.

1.4. Estructura de la tesis

A continuación se realizara una breve reseña de como se estructura el siguiente trabajo:

El capitulo 2 abarca la descripción del problema, en el se define el objetivo general, específicos, hipótesis y los alcances que contiene el proyecto de titulo. Incluyendo también la descripción de la metodología de trabajo utilizada para el progreso del trabajo.

El capitulo 3 posee el marco teórico, en el cual se definen diferentes conceptos esenciales para el entendimiento del proyecto del titulo. Dicho lo anterior se encuentran los siguiente conceptos: Redes centradas en la información (ICN), ndnSIM, aplicaciones web de gran escala, el caché, estructura y políticas de remplazo del caché existentes y finalizando con el comportamiento del usuario de manera donde se explica la ley de Zipf y se detallan los tipos de peticiones de usuario.

En el capítulo 4 se presenta una revisión bibliográfica de los trabajos realizados respecto a las arquitecturas caché y las políticas de caché implementadas en redes ICN, ademas del revisión de trabajos dentro del simulador ndnSIM, todo esto con el fin de diseñar una arquitectura de caché que no exista en la literatura.

Capítulo 2

Descripción del problema

2.1. Contextualización

-¿Que es el caché y para que sirve?

2.2. El problema

2.2.1. Declaración del problema

2.2.2. Diagrama de ishikawa

2.3. Objetivos

2.3.1. Objetivo general

Optimizar la eficiencia del caché en redes *ICN*, por medio del diseño de una arquitectura caché basada en el comportamiento de los usuarios, dentro del simulador *ndnSIM*.

2.3.2. Objetivos específicos

- Determinar los componentes estructurales de las redes *ICN*.
- Proponer una arquitectura caché para redes *ICN* que considere el comportamiento usuario.
- Incorporar una arquitectura caché en el simulador *ndnSIM*
- Comparar el rendimiento de la estrategia caché propuesta con estrategias de la literatura

2.3.3. Alcance de los objetivos

Capítulo 3

Marco teórico

3.1. Information Centric Network

ICN (Information-Centric Networking) es un paradigma que pretende motivar la transición arquitectónica de la actual arquitectura centrada en el host a centrada en la información para difundir de manera eficiente y flexible la enorme información generada por una variedad de aplicaciones. En esta dirección de investigación, se han propuesto muchos enfoques durante los últimos años, de los cuales para el desarrollo de este trabajo nos centraremos en NDN.(Figura 3.1)

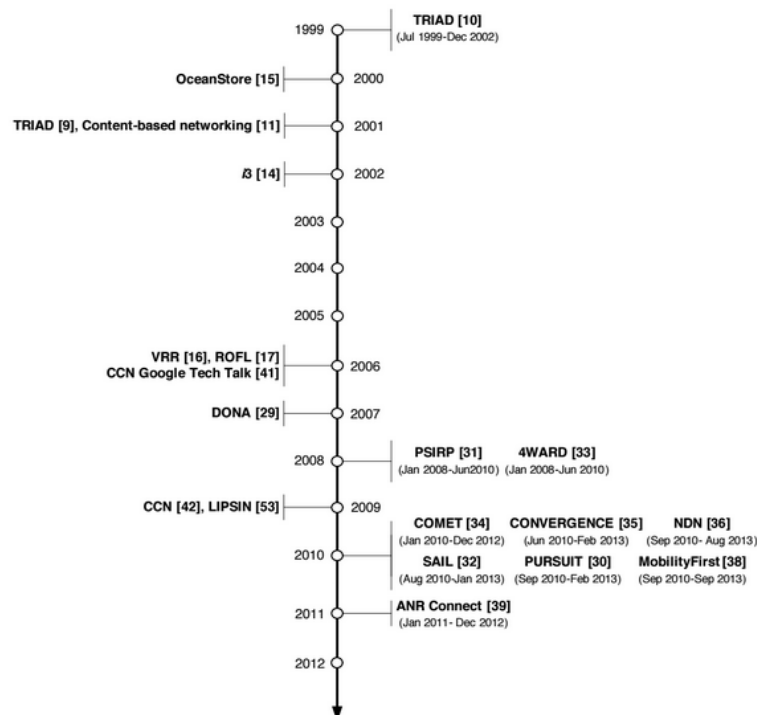


Figura 3.1: Cronología de los principales hitos de ICN [25].

Named Data Networking (NDN) es una de las arquitecturas futuras para el internet con años de investigación sobre el uso de la red y de los problemas no resueltos en arquitecturas contemporáneas del internet como IP[2][1]. La arquitectura NDN tiene su origen en el proyecto Content-Centric Networking (CCN) que Van Jacobson divulgo por primera vez en el año 2006 en una charla tecnológica de Google [12].

El proyecto NDN financiado por EE.UU desarrolla aun más la arquitectura CCN anteriormente mencionada, reconfigurando la pila de protocolos del internet haciendo el intercambio en la cintura delgada de la arquitectura de internet por el de datos nombrados y empleando diversas tecnologías de red por debajo de la cintura para la conectividad, incluyendo IP. También, NDN con el fin de optimizar el uso de recursos, contiene una capa de estrategia media que se ubica entre la capa de datos nombrada y las tecnologías de red, mientras que una capa de seguridad lleva a cabo las funcionalidad de seguridad directamente a los datos nombrados [25] .(Figura 3.2)

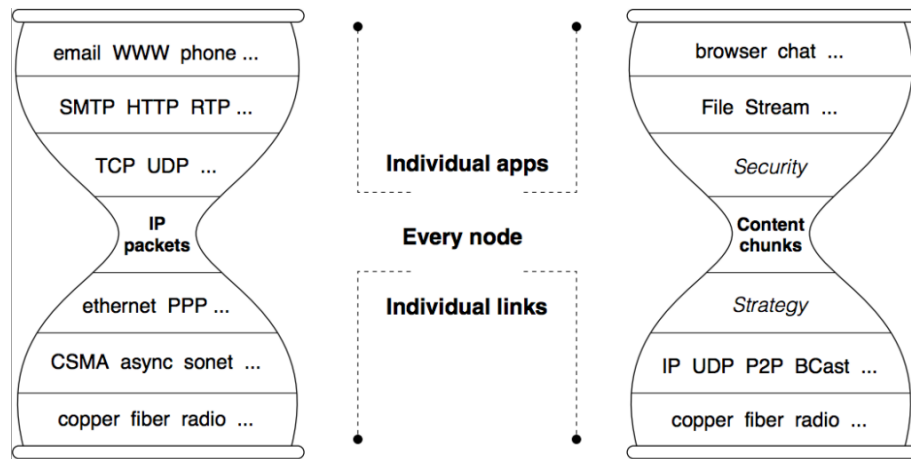


Figura 3.2: Arquitecturas de reloj de arena de Internet y NDN [3].

Dicho lo anterior, la principal abstracción de ICN es el NDO (Named Data Objects), la que se define como la unidad de datos direccionales que se encuentra dentro de una red centrada en la información, siendo capaces de representar una colección de bytes o una información en específica. Así mismo cada uno de los objetos de datos existentes posee un nombre vinculado a el, pudiendo verse como fragmentos de datos nombrados sin semántica alguna caracterizándose por su granularidad la cual varia desde el tamaño del paquete hasta los objetos completos [15][5].

Los objetos de datos con nombre se desligan de la ubicación, la aplicación, el almacenamiento y los medios de los transporte, dando paso al almacenamiento en caché y un bajo costo y ubicua replicación en la red, esperando como resultado una mayor eficiencia y seguridad, una mayor escalabilidad respecto a la demanda de información/ancho de banda y una superior robustez en escenarios de comunicación desafiantes [15].

La arquitectura NDN se compone de dos tipos de NDO, los paquetes de interés(Interest Packet) y los paquetes de datos(Data Packet)(Figura 3.3), siendo el nombre de estos de tipo jerárquico, similares a una URL, como por ejemplo /aueb.gr/ai/main.html.

los cuales son enviados por dos tipos de usuarios el primero denominado consumidor,entidad encargada de enviar una solicitud de un NDO a la red(Interest) y por otra parte un productor, que es la entidad encargada publicar NDO's en la red, los cuales serán solicitados por un consumidor y enviados (Data).

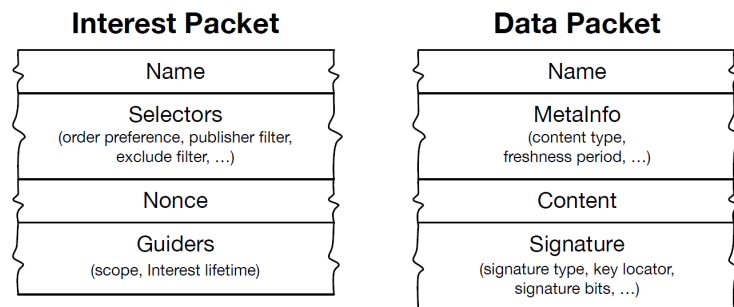


Figura 3.3: Paquetes NDN

Adicionalmente esta arquitectura también cuenta con nodos intermediarios entre los consumidores y los productores denominados CR (Content Routers), los cuales le entregan a la red un aumento en la replicación de información y consigo una mayor resolución de consultas. Los nodos contienen tres estructuras de datos principales(Figura 3.4):

- **Forwarding Information Base (FIB):** El FIB se utiliza para reenviar paquetes de interés hacia fuentes potenciales de datos coincidentes. Es casi idéntico a un IP FIB, excepto que permite una lista de caras salientes en lugar de una sola. Esto refleja el hecho de que CCN no está restringido a reenvío en un árbol de expansión. Permite múltiples fuentes de datos y puede consultarlas todas en

paralelo [13].

- **Almacen de contenido (Content Store (CS)):** El almacenamiento en caché de los NDO es una parte integral del servicio ICN. Todos los nodos potencialmente tienen cachés, incluyendo nodos en redes de infraestructura de ejecución de operadores y redes domésticas dirigidas por usuarios, así como terminales móviles. Las solicitudes de NDO pueden ser satisfechas por cualquier nodo que contenga una copia en su caché. ICN combina así el almacenamiento en caché en el borde de la red, como en P2P y otras redes de superposición, con almacenamiento en caché en la red (por ejemplo, cachés de web transparentes). El almacenamiento en caché es genérico, es decir, es independiente de la aplicación y se aplica a todos los proveedores de contenido, incluido el contenido generado por el usuario [13].
- **Pending Interest Table (PIT):** El PIT realiza un seguimiento de los Intereses enviados hacia arriba hacia la(s) fuente(s) de contenido de forma que los datos devueltos puedan ser enviados aguas abajo a su(s) solicitante(s) [13].

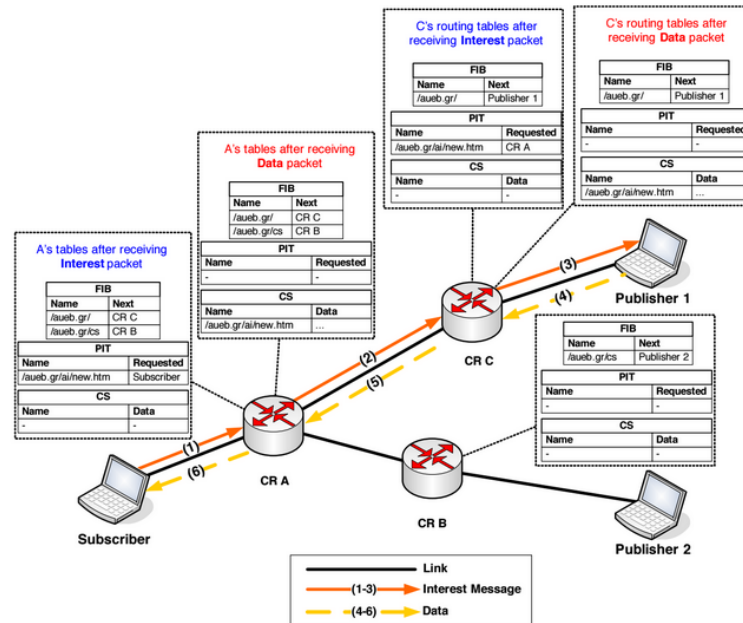


Figura 3.4: Nodo CCN

3.2. Caché

La cantidad de usuarios van en aumento día a día, por lo que las aplicaciones web con un gran numero de usuarios tienen que ser capaces de soportar grandes volúmenes de consultas y transacciones de datos(millones), las cuales poseen una patrón de comportamiento definido, este tema se abordara en (). Una de las tecnologías que existe para apaciguar el acceso incesante a información de los servidores, es la de un lugar de almacenamiento destinado a archivar las consultas y su respuesta pre-computada, denominado *caché*.

El *caché* es una de las partes mas importante dentro de los servicios web y motores de búsqueda [6],[4]. Principalmente el caché es un espacio de almacenamiento(memoria) de acceso rápido, en donde se retienen las consultas más reiteradas hechas por los usuarios y sus respuestas, entregando una gran cantidad de ventajas entre las que se encuentran:

- Reducir la latencia de acceso a los documentos, esto se debe a que los objetos que son accedidos con mayor frecuencia se encuentran almacenados en un nodo de caché cercano, lo que disminuye el tráfico en la red, permitiendo que los objetos que no están en caché puedan también ser recuperados con mayor rapidez [22].
- Reducir el consumo de ancho de banda, lo que permite disminuir la congestión y el tráfico en la red [22].
- Reducir la carga de trabajo en los servidores web [22].

Siendo las características codiciables para una arquitectura caché las siguientes: robustez(capaz de no generar fallos o bloquearse), acceso rápido (disminuir latencia), eficiente, escalable, estable, adaptable y simple de implementar [22]. Por ello una de las finalidades más importante en el *caché* es la de obtener el mayor numero respecto a la tasa de hit, en otras palabra, conseguir que una gran parte de las consultas realizadas por el usuario, que ingresan al caché cuente con una respuesta pre-computada.

En la arquitectura NDN, el caché de los CR juega un rol importante al momento de satisfacer las peticiones del usuario(Consumidor). A modo de que sin la presencia del caché, el paradigma de las redes ICN no seria valida, causando así que los productores(Servidores), procesen un volumen elevado de consultas simultaneas.La importancia del caché en la arquitectura NDN de las redes ICN radica en que esta requiere de una difuminación eficiente de la gran cantidad de consultas realizadas

diariamente por lo usuarios, de manera que sean respondidas de forma rapida.

El caché es limitado, por lo que al completar su capacidad máxima es necesaria la implementación de algoritmos que permitan el control del almacenamiento, estos algoritmos son las denominadas políticas, de las cuales existen tanto de admisión, reemplazo y desalojo. Dada la gran cantidad de consultas que generan los usuarios en el internet es que se hace necesaria la creación de nuevas políticas que permitan una mejor eficiencia del caché.

3.2.1. Organización del Cache

En la arquitectura NDN el *caché* (*Content Store*) se encuentra presente en los distintos nodos (CR) incluyendo también a los consumidores y productores. Donde la entrada que posee un CS es una tupla ¡Name, Data¿siendo Name el NDO Interest y Data el NDO que responde dicho Interest. Como se señalo en el punto anterior, el almacenamiento en caché es limitado, por lo que la entrada de tuplas deben ser constantemente variadas, dando la posibilidad del ingreso de nuevas entradas, las cuales tienen mayor probabilidad de aumentar la tasa de hit en caché. A continuación se definirán las políticas ya señaladas

3.2.1.1. Política Admisión

Esta política es la encargada del administrar la entrada a la memoria caché, a través del bloqueo de entradas que no sean constantemente requeridas en el futuro, a raíz de que lo anterior producirá un aumento en la tasa de miss del caché [7], disminuyendo la eficiencia del nodo CR.

Esto ocurre cuando la memoria en caché se encuentra en su máxima capacidad, por lo que no es posible almacenar una nueva entrada, motivo por el cual se pone en marcha la política de desalojo respectiva para descartar una entrada del caché y ingresar la entrada entrante. Pero debido a que la nueva entrada puede generar escasos hit's entonces se deben de llevar a cabo las políticas de admisión las cuales tomaran la decisión de si la nueva entrada es guardad en la memoria caché o bien se mantendrá la entrada propuesta para el desalojo de caché.

3.2.1.2. Política Desalojo o Reemplazo

Las política de desalojo o reemplazo se llevan a cabo cuando la memoria caché se encuentra llena, por lo que es necesario liberar espacio para dar oportunidad a otras entradas entrantes, principalmente la idea es descartar una entrada la cual posee una baja probabilidad de lograr ser un hit en caché [7].

Los objetos web, los NDO en el caso de la arquitectura NDN contienen diferentes variables que son tomadas en cuenta para decidir si salen o se quedan en caché, como lo son el tiempo de la ultima referencia, el tamaño, tiempo de expiración y el tiempo desde su ultima modificación.

Algunas de políticas de desalojo conocidas dentro de la literatura, son *Least Frequently Used*(LFU), *Least Recently Used* (LRU) [11][10][9]. Dentro del simulador ndnSIM encontraremos las ya anteriormente señalados algoritmos de reemplazo, además de el algoritmo FIFO, y un cantidad no menor de variaciones de los algoritmos señalados, con su respectiva variable que permitirá decidir que entrada seleccionar.

3.2.1.3. Eliminación de entradas antiguas

Principalmente la política de eliminación de entradas antiguas son las encargadas de mantener el almacenamiento en memoria caché actualizado.

3.2.2. Estructura del caché

Anteriormente vimos que es posible caracterizar el caché según la políticas que utiliza este mismo, pero también por otro lado es factible agrupar el caché en dos categorías: Estático y dinámico.

3.2.2.1. Caché Estático

Tal como dice su nombre, son cachés que no cuentan con variabilidad de contenido durante un intervalo de tiempo grande, estos tipos de caché se basan generalmente en información acumulada históricamente y es actualizada habitualmente.

Dentro del contexto de los motores de búsqueda, el caché estático aglomera en memoria las entradas más habituales, por medio del análisis de registros históricos de las entradas efectuada por los usuarios. Siendo algunas de las características a tomar en cuenta para ser incorporada en el caché: la cantidad de usuarios que realizaron la misma consulta, la frecuencia de la consulta, proporción de clics hechos por los usuarios a las URL que entregaron como respuesta a la consulta, además de la estabilidad temporal de la frecuencia de una consulta.

Realizada una vez la captación de las entradas mas habituales, serán guardadas en caché y actualizada oportunamente.

3.2.2.2. Caché Dinámico

Son cachés en donde el contenido varia constantemente, por lo tanto se trata de incorporar dentro del caché las entradas que tienen alta probabilidad de ser solicitadas en un futuro. Por lo tanto al ser una caché que varia en el tiempo, el contenido almacenado debe ser frecuentemente actualizado, por medio de políticas de desalojo o reemplazo. Inicialmente este tipo de caché esta vacía, para luego ir llenándose periódicamente a medida que se realizan entradas en caché (tuplas) hasta llenarse y luego implementar las políticas de reemplazo para mantenerse actualizado.

3.3. Comportamiento de usuario

El comportamiento de usuario se puede definir como, un área que intenta encontrar características, patrones o conductas de los usuarios en algún contexto específico, como por ejemplo en las redes sociales, en algunos sistemas de compras, y otras. Esto es de importancia para las empresas en general, ya que pueden mejorar sus ofertas de productos o hacer sistemas más eficientes que consideren alguna conducta repetitiva de los usuarios.

3.3.1. Ley de Zipf

La llamada ley de Zipf, formulada en la década de 1940 por George Kingsley Zipf [27], [28] es una ley empírica la cual establece que una cantidad menor de palabras son altamente utilizadas, mientras que por otro lado existe un conjunto mayoritario

de palabras que son raramente usadas. El comportamiento o bien patron que tienen estas palabras siguen la formula:

$$\frac{1}{n} \tag{3.1}$$

Donde n corresponde a la n-ésima palabra ordenada por frecuencia decreciente. El comportamiento anteriormente señalado no solo ha sido estudiado para las palabras, también a sido analizado en el contexto de las WSE (Motores de búsqueda), específicamente al comportamiento de las consultas, obteniendo como resultado que estas también siguen una distribución "zipfianas" [8], [9], [10], [16], [23], [21], [24].

3.3.2. Peticiones de usuarios

Los usuarios del Internet de ahora ya no son los mismos usuarios del internet de años atrás, la información que necesitaban los primero usuarios, ya no es la misma información que necesitan los usuarios de ahora y al igual que la moda, esta va variando en el tiempo. De la misma manera las consultas generadas por los usuarios va cambiando en el tiempo, en función de distintas circunstancias que afectan el interés de los usuarios a la hora de realizar consultas, como lo son la economía, el deporte, desastres naturales, fechas importantes, temporada, eventos sociales y/o culturales, etc.

Teniendo en cuenta lo anterior y la frecuencia con la que se lleva a cabo cierto tipo de consulta, es que se a logrado encontrar patrones de conducta los cuales nos permite clasificar los distinto tipos de consultas. Entre las clasificaciones de consultas más conocidas se encuentran las consultas periódicas, las consultas en ráfaga y las consultas permanentes, las que serán detalladas a continuación.

3.3.2.1. Consultas Periódicas

Corresponden a consultas que poseen un aumento en su frecuencia en un intervalo definido de tiempo, para luego disminuir hasta llegar a su frecuencia constante, todo esto de manera periódica(Meses, semanas, días, horas, años, etc.). Dicha información permite anticipar la sobrecarga de lo servidores, sin tener la necesidad de utilizar un

algoritmo que permita la admisión en caché. A continuación en la figura se muestra un ejemplo de una consulta de tipo periódica, correspondiente a la consulta "18 de septiembre"



Figura 3.5: Frecuencia de consulta periódica: "18 de septiembre (2004-2016)". Fuente: Google Trends.

3.3.2.2. Consultas Ráfagas

Este tipo de consultas se caracteriza principalmente por poseer una frecuencia constante en el tiempo, la cual se ve afectada por un evento en específico ya sea de índole social, económico, cultural, etc. llamando así el interés de los usuarios en general. Provocando que la frecuencia respecto a un tema en específico aumente bruscamente hasta alcanzar un pick, para luego ir reduciéndose a medida que pasa el tiempo, como consecuencia del desinterés que comienza a tener los usuarios respecto al tema mencionado. Un ejemplo de este tipo de consultas corresponde a la "muerte de Fidel Castro" tal como se muestra en la Figura 3.6.

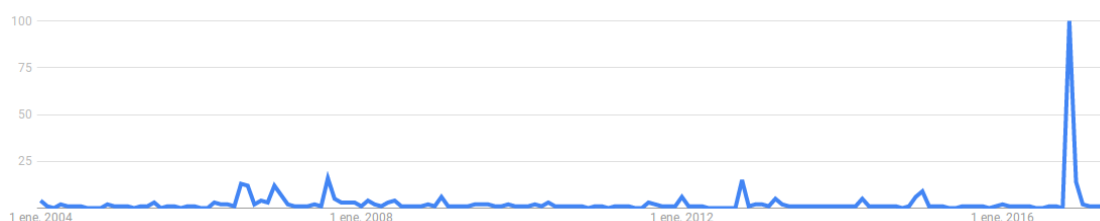


Figura 3.6: Frecuencia de consulta ráfaga: "Muerte de Fidel Castro"(2004-2017). Fuente: Google Trends.

Ademas, las consultas en ráfaga no solo afectan a una consulta en específico, si no que también a todo conjunto de consultas que tengan algún tipo de relación con el evento específico que llevo al alza en este topico. En la Figura 3.7 se puede observar

como también la consulta "Historia de Fidel Castro" posee un patrón ráfaga de la misma forma que la consulta "muerte de Fidel Castro" durante los mismos intervalos de tiempo.

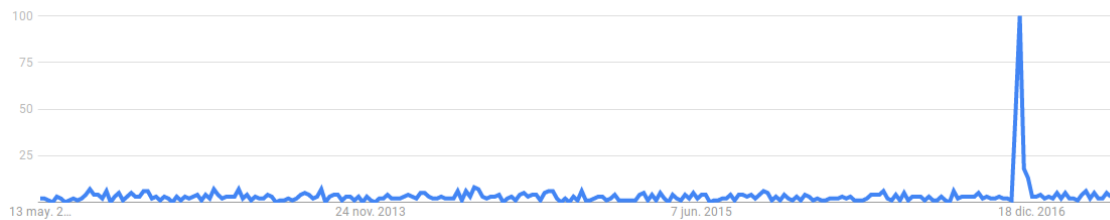


Figura 3.7: Frecuencia de consulta rafaga: "Historia de Fidel Castro"(2012-2017). Fuente: Google Trends.

Uno de los problemas mas serios que producen estas consultas debido al aumento en el volumen de consultas solicitadas, es la generación de desbalance en los nodos con respuestas pre-computadas en caché, como también una congestión al acceso de la información en los servidores.

3.3.2.3. Consultas Permanentes

Finalmente las consultas permanentes, son definidas como aquellas que poseen una continuidad en la frecuencia que son solicitadas en intervalos largo de tiempo, en simples palabras son consultas que el paso del tiempo no afecta abruptamente su frecuencia. En la Figura 3.8 se adjunta una consulta del tipo permanente la cual corresponde a "YouTube".

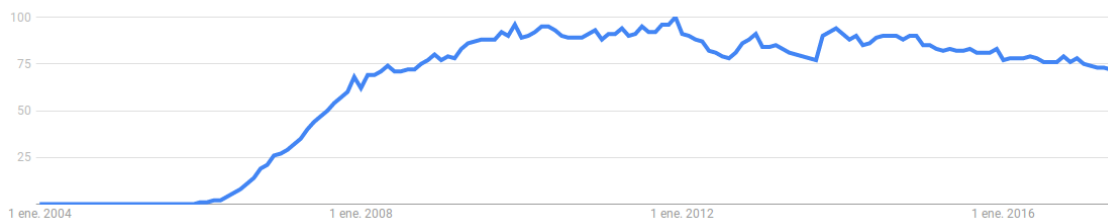


Figura 3.8: Frecuencia de consulta permanente: "YouTube"(2004-2017). Fuente: Google Trends.

Bajo este contexto, muchos son los trabajos encontrados en la literatura [26], [17], [19], [14], los cuales tienen como conclusión que destinar un espacio estático en caché para guardar las consultas permanentes, lo que permite aumentar el rendimiento del caché, mejorando la tasa hit de este mismo. Cabe recordar que las consultas permanente se obtienen de registros históricos, los cuales son analizados para determinar cuales son consultas permanente y cuales no, tomando en cuenta su frecuencia.

Capítulo 4

Estado del arte

4.1. NDNSIM

4.1.1. On Performance of Cache Policies in Named Data Networking [20]

En [20] se propone el diseño de una política de reemplazo de caché basada en la popularidad del contenido (CCP), la cual tiene en cuenta el factor de popularidad y una nueva estructura de datos denominada "Tabla de popularidad de contenido (CPT) sobre la base de retención del Content Store (CS), Pending Interest Table (PIT) y la Forwarding Interest Base (FIB). Con el fin de beneficiar las métricas de golpe de caché, carga del servidor y el promedio de rendimiento de la red en comparación con las políticas de caché LFU y LRU ya presentes en el simulador.

Para que esto sea posible, según [20] las estrategias caché deben poseer dinamismo en cuanto al contenido almacenado en caché, siguiendo los dos siguientes puntos:

- Ser capaz de adaptarse a los cambios rápidos en el modo de transmisión dinámica de red.
- No puede ser afectado por el contenido de la alternativa temporal.

De modo que las demandas anteriormente señaladas son saciadas por la política de reemplazo de caché basada de popularidad (CPP) propuesta.

4.1.1.1. Calculo de la popularidad de un contenido

CCP añade una nueva estructura de datos CPT, la cual contiene toda la información sobre la popularidad del contenido almacenada en CS, incluyendo el nombre

del contenido, el hit en caché, la popularidad anterior y actual. Dicha estructura de datos se va actualizando cada vez que termina un ciclo de conteo(PREGUNTAR), obteniendo así el ranking actual de popularidad del contenido por medio del hit de caché y la popularidad anterior. A continuación se describe la formula de calculo para la popularidad del contenido:

$$P[i + 1] = \frac{N[i] * \alpha + P[i]}{\alpha + 1} \quad (4.1)$$

$$\alpha = 1 + c * T \quad (4.2)$$

En donde $P[i]$ corresponde a la popularidad del contenido en caché y $N[i]$ los hits del contenido en caché en el momento del conteo actual. En cuanto a α ($\alpha > 1$) es el coeficiente de peso de la popularidad del contenido, T el ciclo de conteo y C el coeficiente proporcional de α y T . Señalado lo anterior, para este trabajo desarrollado se considero que $c = 0,5$ y $T = 8$.

Pero para calcular la popularidad de un contenido, no solo se toma en cuenta el peso entre la popularidad anterior y actual, si no que también se considera la influencia de la ultima red hot pot. Logrando así por medio de la expansión de 4.1 la obtención de 4.3.

$$P[i + 1] = \frac{N[i] * \alpha + P[i]}{\alpha + 1} = \frac{N[i] * \alpha}{\alpha + 1} + \frac{N[i - 1] * \alpha}{(\alpha + 1)^2} + \frac{N[i - 2] * \alpha}{(\alpha + 1)^3} + \dots \quad (4.3)$$

Llegando así a la conclusión de que con el tiempo, el numero de hits anterior posee una influencia menor en la popularidad de l contenido, lo que refleja de mejor manera la popularidad del contenido en la red actual. Obteniendo resultados como los del siguiente ejemplo, pudiéndose ver que el segundo contenido tiene una popularidad anterior mucho mayor que la primera pero con un menor caché actual.

Content name	Previous popularity	Current cache hit	Current popularity
/campus/images/v1.mpg	12.08	20	18.68
/campus/images/v2.mpg	21.28	15	16.05
/campus/music/m1.mp3	22.14	30	28.69

Figura 4.1: Tabla de popularidad de contenido. Fuente: On Performance of Cache Policies in Named Data Networking [20]

4.1.1.2. Estrategia de reemplazo

La estrategia de reemplazo del presente trabajo consta de calcular la diferencia entre el valor de umbral del tamaño de caché y el número de contenido en caché, obteniendo así el tamaño de espacio de caché restante. Por lo tanto si un paquete de datos es mayor que el resto del espacio, el reemplazo ocurrirá, siendo el contenido con la menor popularidad el seleccionado para su reemplazo y eliminando también su registro en CPT. Para que luego el nuevo contenido llegado, se almacene en caché se registre el CPT al mismo tiempo.

4.1.1.3. Simulación y evaluación

Para lograr cuantificar la efectividad de CCP se utilizó el paquete ndnSIM de código abierto, que implementa la pila de protocolos NDN para el simulador de red NS-3 para ejecutar simulaciones. Se implementará CCP dentro de ndnSIM para evaluar la efectividad de CCP respecto a otras estrategias presentes en el simulador (LFU, LRU). Las métricas utilizadas para cuantificar el rendimiento son la proporción de cache hit, el promedio de rendimiento de la red y carga del servidor. Se asume que los usuarios expresan intereses según una tarifa de promedio constante con un intervalo de tiempo aleatorio entre dos intereses consecutivos según una distribución uniforme.

Para la simulación se utilizaron 20 routers de contenido, los usuarios se distribuyen en el borde de la red y solo existe un servidor para satisfacer los intereses solicitados. En la Fig 4.2 se puede observar parte de la topología utilizada. Se utilizaron 200 objetos de contenidos diferentes en la red. La capacidad del cache equivale a un porcentaje que oscila entre el 20 % y 60 % de la cantidad total de objetos de contenido en la red y el tiempo definido para la simulación es de 300s. Además de asumir un ancho de banda de backhaul(PREGUNTAR) de 100Mbps y un tamaño de cache que varía entre 20 y 400Kbits.

4.1.1.4. Resultados

A continuación se presentarán los resultados obtenidos respecto a las pruebas efectuadas a CCP en el simulador ndnSIM.

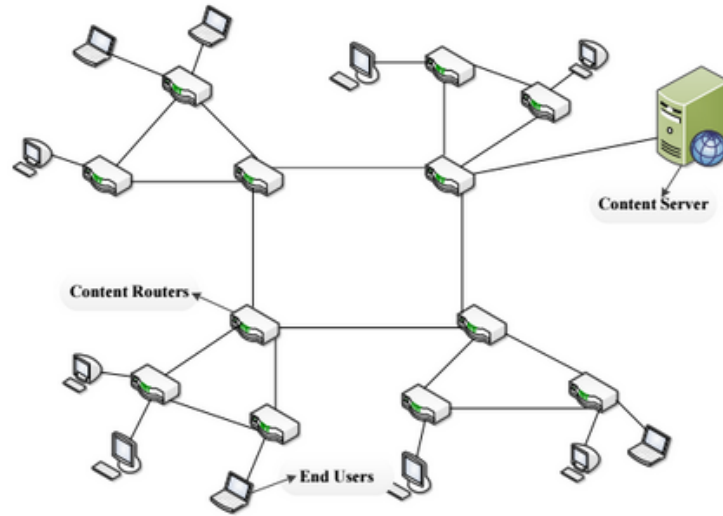


Figura 4.2: Sección de la topología de la red. Fuente: On Performance of Cache Policies in Named Data Networking [20]

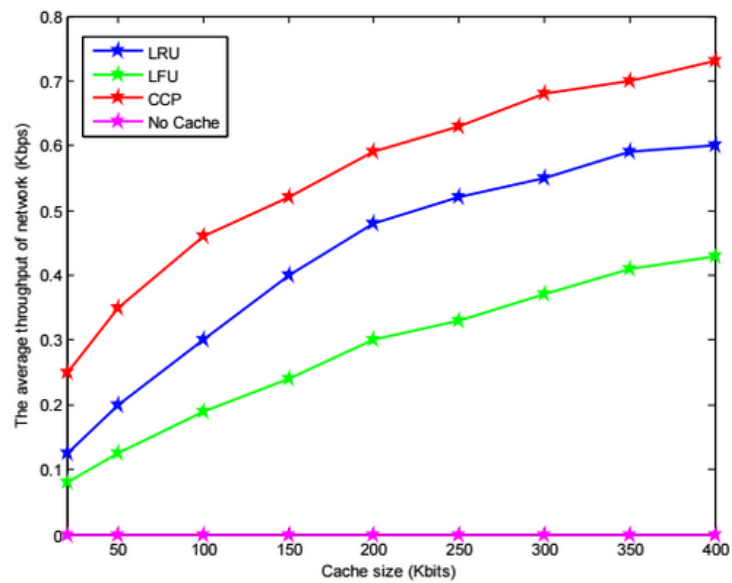


Figura 4.3: Proporción de hit en caché VS Tamaño caché. Fuente: On Performance of Cache Policies in Named Data Networking [20]

4.2. Arquitectura de Caches

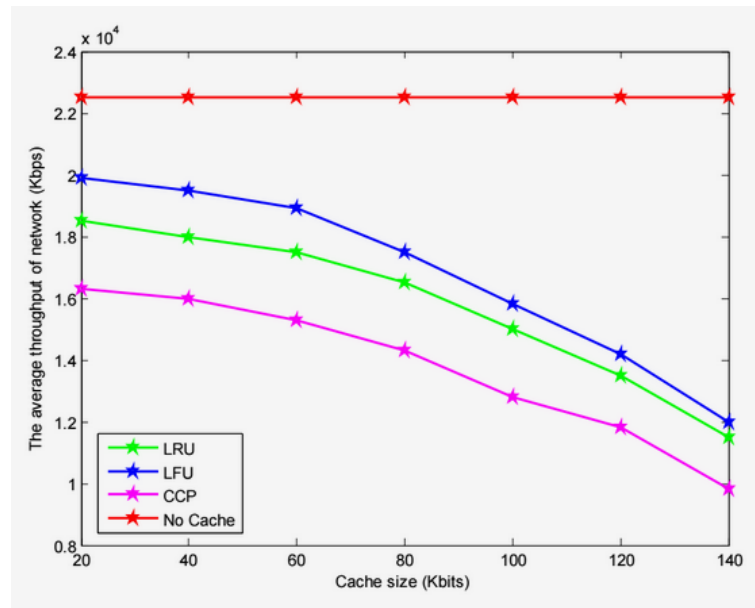


Figura 4.4: Rendimiento medio de la red VS Tamaño caché. Fuente: On Performance of Cache Policies in Named Data Networking [20]

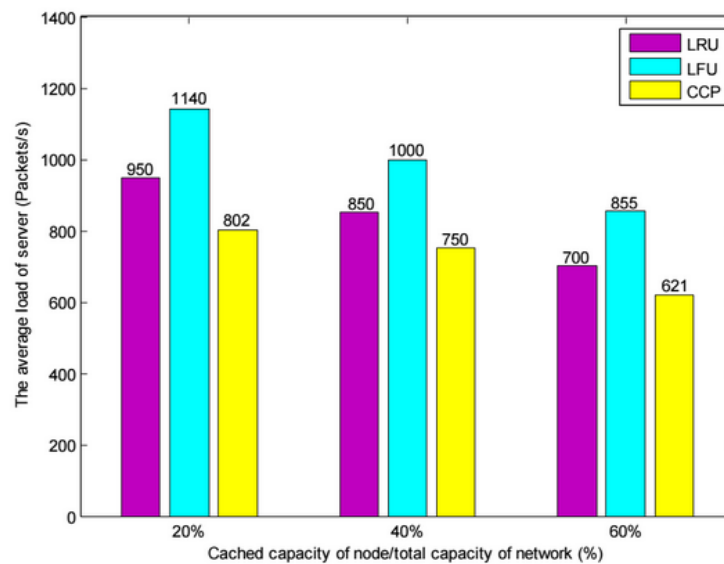


Figura 4.5: Carga media del servidor VS Capacidad caché. Fuente: On Performance of Cache Policies in Named Data Networking [20]

Bibliografia

- [1] Future internet architectures – next phase (fia-np) (nsf13538).
<https://www.nsf.gov/pubs/2013/nsf13538/nsf13538.htm>.
- [2] Future internet architectures (fia) nsf10528.
<https://www.nsf.gov/pubs/2010/nsf10528/nsf10528.htm>.
- [3] Named data networking: Executive summary - named data networking (ndn).
<https://named-data.net/project/execsummary/>.
- [4] Charu Aggarwal, Joel L Wolf, and Philip S. Yu. Caching on the world wide web. *IEEE Transactions on Knowledge and data Engineering*, 11(1):94–107, 1999.
- [5] Bengt Ahlgren, Christian Dannewitz, Claudio Imbrenda, Dirk Kutscher, and Borje Ohlman. A survey of information-centric networking. *IEEE Communications Magazine*, 50(7), 2012.
- [6] Ismail Sengor Altingovde, Rifat Ozcan, and Özgür Ulusoy. A cost-aware strategy for query result caching in web search engines. In *European Conference on Information Retrieval*, pages 628–636. Springer, 2009.
- [7] Ricardo Baeza-Yate, Flavio Junqueira, Vassilis Plachouras, and Hans Friedrich Witschel. Admission policies for caches of search engine results. In *International Symposium on String Processing and Information Retrieval*, pages 74–85. Springer, 2007.
- [8] Ricardo Baeza-Yates and Simon Jonassen. Modeling static caching in web search engines. In *European Conference on Information Retrieval*, pages 436–446. Springer, 2012.
- [9] Berkant Barla Cambazoglu, Flavio P Junqueira, Vassilis Plachouras, Scott Banachowski, Baoqiu Cui, Swee Lim, and Bill Bridge. A refreshing perspective of search engine caching. In *Proceedings of the 19th international conference on World wide web*, pages 181–190. ACM, 2010.

- [10] Qingqing Gan and Torsten Suel. Improved techniques for result caching in web search engines. In *Proceedings of the 18th international conference on World wide web*, pages 431–440. ACM, 2009.
- [11] Carlos Luis Gómez Pantoja. Servicios de cache distribuidos para motores de búsqueda web. 2014.
- [12] Van Jacobson. A new way to look at networking. *Google Tech Talk*, 30, 2006.
- [13] Van Jacobson, Diana K Smetters, James D Thornton, Michael F Plass, Nicholas H Briggs, and Rebecca L Braynard. Networking named content. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pages 1–12. ACM, 2009.
- [14] Chetan Kumar and John B Norris. A new approach for a proxy-level web caching mechanism. *Decision Support Systems*, 46(1):52–60, 2008.
- [15] Dirk Kutscher, Suyong Eum, Kostas Pentikousis, Ioannis Psaras, Daniel Corujo, Damien Saucez, T Schmidt, and Matthias Waehlich. Information-centric networking (icn) research challenges. Technical report, 2016.
- [16] Ronny Lempel and Shlomo Moran. Predictive caching and prefetching of query results in search engines. In *Proceedings of the 12th international conference on World Wide Web*, pages 19–28. ACM, 2003.
- [17] Evangelos P. Markatos. On caching search engine query results. *Computer Communications*, 24(2):137–143, 2001.
- [18] Paul Müller and Bernd Reuther. Future internet architecture—a service oriented approachfuture internet architecture—ein serviceorientierter ansatz. *it-Information Technology*, 50(6):383–389, 2009.
- [19] Rifat Ozcan, I Sengor Altingovde, B Barla Cambazoglu, Flavio P Junqueira, and ÖZgür Ulusoy. A five-level static cache architecture for web search engines. *Information Processing & Management*, 48(5):828–840, 2012.
- [20] Jianhua Ran, Na Lv, Ding Zhang, Yuanyuan Ma, and Zhenyong Xie. On performance of cache policies in named data networking. In *International Conference on Advanced Computer Science and Electronics Information*, pages 668–671, 2013.

- [21] Paricia Correia Saraiva, Edleno Silva de Moura, Novio Ziviani, Wagner Meira, Rodrigo Fonseca, and Berthier Riberio-Neto. Rank-preserving two-level caching for scalable search engines. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 51–58. ACM, 2001.
- [22] Jia Wang. A survey of web caching schemes for the internet. *ACM SIGCOMM Computer Communication Review*, 29(5):36–46, 1999.
- [23] Jianguo Wang, Eric Lo, Man Lung Yiu, Jiancong Tong, Gang Wang, and Xiaoguang Liu. The impact of solid state drive on search engine cache management. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 693–702. ACM, 2013.
- [24] Yinglian Xie and David O’Hallaron. Locality in search engine queries and its implications for caching. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1238–1247. IEEE, 2002.
- [25] George Xylomenos, Christopher N Ververidis, Vasilios A Siris, Nikos Fotiou, Christos Tsilopoulos, Xenofon Vasilakos, Konstantinos V Katsaros, and George C Polyzos. A survey of information-centric networking research. *IEEE Communications Surveys & Tutorials*, 16(2):1024–1049, 2014.
- [26] Weizhe Zhang, Hui He, and Jianwei Ye. A two-level cache for distributed information retrieval in search engines. *The Scientific World Journal*, 2013, 2013.
- [27] George Kingsley Zipf. *The psycho-biology of language: An introduction to dynamic philology*. Routledge, 2013.
- [28] George Kingsley Zipf. *Human behavior and the principle of least effort: An introduction to human ecology*. Ravenio Books, 2016.