# IPFS: Interplanetary file system

Departamento de Electrónica, Universidad Técnica Federico Santa María
Matías Barrios 201121027-1
Felipe Cordero 2903012-K
Nicolás Hernández 201121020-4

## I. Resumen

IPFS es un sistema de archivos distribuidos que busca conectar todos los computadores con el mismo sistema de archivos. Entre las principales motivaciones para el desarrollo de este protocolo, esta la volatilidad actual de la información en la red. Mediante un sistema P2P, los usuarios pueden albergar información en IPFS y acceder a ella sin la necesidad de que exista un único punto de acceso. IPFS utiliza un sistema de direccionamiento por contenido, a diferencia de HTTP que direcciona mediante ubicación, lo que permite generar beneficios en términos de calidad de servicio.
Entre las razones para hacer esta investigación acerca de este protocolo, está la corriente actual acerca de la tendencia a los sistemas distribuidos. Ejemplos como BitTorrent o Blockchain [9] han dejado en claro que la descentralización de servicios trae muchos beneficios, entre ellos la clara oportunidad de no depender de la confianza en un tercero para la autenticación de información y datos. .

## II. Motivación y problemas planteados

### A. *HTTP*

En la actualidad, el protocolo para intercambio de información por excelencia es HTTP. En este, el contenido es accedido mediante su ubicación, a traves de una url que lleve a acceder a la información requerida. Esto produce diversos problemas:

- Debido a que el contenido se encuentra en un único punto accedido mediante URL, los servidores deben estar capacitados para recibir altos niveles de tráfico, lo que en ocasiones puede generar problemas de latencia debido a sobrecarga en la conexión al servidor.
- El acceso a contenido mediante ubicación genera en el sistema un único punto de falla, por lo que de fallar este, la información no sería accesible para quienes la deseen.
- La alta volatilidad de la información en la red genera que información se pierda para siempre, generando inconsistencias en los direccionamientos por ubicación.

IPFS busca ser la solución a este problema, mediante la propuesta de un sistema de *file-sharing*, combinando tecnologías como *Git* [6] o BitTorrent. Mediante el uso de un *Merkle Tree*, se logra direccionar a los usuarios a acceder a información mediante direccionamiento en contenido y no en ubicación. IPFS fue diseñado para ser usado de múltiples formas:

- Como un sistema de archivos global, usando /ipfs y /ipns.
- Como un archivo de actualización automática de versionas, publicaciones y resplados.
- Como un sistema de *file-sharing*.
- Como un administrador de versiones de paquete para software.
- como una plataforma conectada y encriptada de comunicación.
- Como un CDN con integridad comprobada para archivos grandes.
- Como un CDN encriptado
- Como una red permanente donde los links no mueren.

IPFS plantea como objetivo de implementación:

- una librería IPFS para importar a desarrollo de aplicaciones.
- linea de comandos para manipular objetos directamente.
- como un metodo de almacenamiento digital distribuido.

## III. Como funciona

### A. Identidades

Los nodos son identificados mediante *NodeId*, el cual es el hash de la llave publica creada con *S/Kademlia's static crypto puzzle* [1]. Cada nodo guarda tanto su llave pública como su llave privada. Cuando un nodo se quiere comunicar con otro, intercambian llaves públicas, y luego de comprobar si es que existen dichas llaves, se establece la conexión, de otro modo, la conexión termina.
Los usuarios pueden crear un nodo cada vez que inician una sesión en IPFS, pero se les incentiva a no usar esta práctica para no perder beneficios en torno a una red de nodos innecesariamente grande. Los hash utilizados para *NodeId.*, cuyo formato es llamado *multihash*, incluye un *header* especificando la función hash utilizada y el largo de la misma. Esto permite al sistema tomar la mejor decisión en torno al compromiso de performance v/s seguridad.

### B. Red

El protocolo de red de IPFS incluye servicios para transporte, confiabilidad, conectividad, integridad y autenticidad [2]:

- Transporte Puede utilizar cualquier protocolo de la capa transporte. Ha sido probado con éxito en WebRTC o $\mu$TP.
- Confiabilidad Puede proveer de confianza usando $\mu$TP o SCTP [5].
- Conectividad Utiliza *ICE NAT traversal techniques* [7]
- Integridad De manera opcional, revisa integridad de mensajes utilizando *hash checksum*.
- Autenticidad De manera opcional, revisa autenticidad de mensajes utilizando *HMAC*[8] con la llave pública del nodo que requiere conexión.

### C. Enrutamiento

Los nodos IPFS requieren un sistema de enrutamiento que permita encontrar direcciones de otros *peers* de la red y *peers* que permitan servir un objeto particular [2]. Lo logra utilizando *DHT based on S/Kademlia* [1] y *Coral*[3].

### D. Intercambio

En IPFS, la distribución de datos ocurre mediante intercambio de bloques entre *peers* utilizando un protocolo basado en BitTorrent: BitSwap [2]. Como en BitTorrent, los *peers* de BitSwap buscan adquirir un serie de bloques (want_list), ofreciendo otro a cambio (have_list).
A diferencia de BitTorrent, BitSwap no busca parte especificas de un archivo, si no que pide bloques particulares, sin importar si su origen es un archivo u otro.
IPFS busca optimizar el método de *file-sharing*:

- Maximizar el performance para cada nodo, y del intercambio completo.
- Prevenir la aparición de *leechers*, o usuarios que no aportan al funcionamiento de la red mediante *file-sharing*.
- Funcionar a favor de *peers* de confianza.

### E. Objetos

*DHT*[4] y *BitSwap* le permiten a IPFS formar un sistema P2P masivo para albergar y distribuir bloques de manera rápida y robusta. IPFS utiliza un sistema similar al utilizado por Git [6], lo que le otorga propiedades útiles como:

- Direccionamiento por contenido: Todo el contenido es identificado únicamente mediante su *multihash checksum*, incluyendo links.
- Resistencia a manipulación ajena: Todo el contenido está verificado con *checksum*. Si la información es alterada, IPFS es capaz de detectarlo.
- No duplicación: Todos los objetos con el mismo contenido son iguales, y solamente almacenados una vez.

El formato de un objeto IPFS es el siguiente:

```
type IPFSLink struct {
Name string
// nombre o alias de link
Hash Multihash
// hash del archivo objetivo
Size int
// t a m a o total
}

type IPFSObject struct {
links []IPFSLink
// arreglo de links
data []byte
// datos ofuscados
}
```

Para IPFS, el único requerimiento es que el objeto a compartir tenga el formato mostrado anteriormente, sin importar el origen de la *data* del mismo objeto, pudiendo ser incluso de algún tipo que IPFS no

pueda interpretar.Esto le permite actuar como un protocolo de transporte para diversos propósitos.

*1) Path:* Los objetos IPFS pueden ser accedidos con un *string path API*[2]. *Paths* funcionan como lo hacen tradicionalmente los sistemas UNIX:

```
/ipfs/XLF4hwVHsVuZ78FZK6fozf8Jj9WEU4/hello.txt
```

Al ser un sistema de archivos distribuidos, no hay un usuario *root* que tenga permisos especiales. Por eso mismo ademas el primer objeto despues de /ipfs/ es un hash de un objeto. En este caso, el hash es el root provisional para ese objeto.

### F. Archivos

Los archivos constan de 4 categorías:

- *blob*: bloque de datos de tamaño variable.
- *list*: colección de bloques o otras listas.
- *tree*: colección de bloques, listas o otros *trees*.
- *commit*: una ”fotografía” de la versión del *tree*

Mediante estas estructuras, y con el formato de Git[6], se logra un control de versiones acerca de los archivos compartidos. IPFS prefiere almacenar la información en forma de *trees*, debido a que la utilización solamente de *blobs* es ineficiente a la hora de hacer búsqueda. Cuando se necesita compartir un nuevo estado de un archivo, se distribuye el *commit* y el *tree* que incluye este cambio en el estado de la red mediante un *flattened tree*.

### G. Nombramiento

Los nombres para cada archivo deben ser únicos, lo que se logra mediante una función *hash*. El *IPFS Merkle DAG*, el cual es el sistema de enlistamiento de archivos para IPFS, permite direccionar mediante estos nombres asociados al contenido del archivo. Si algun usuario intenta compartir un archivo que ya existe en la red, no se volverá a subir, y se le entregará el *hash* perteneciente a dicho archivo. Como un mismo archivo puede ser de interés de múltiples usuarios, es necesario entonces que los archivos compartidos en el sistema **sean permanentes**. De esta forma, al querer compartir un archivo, solo se debe compartir su *hash*, de forma de acceder a el mediante el protocolo de IPFS. Los usuarios pueden incluso hacer direccionamiento directo a los *hash* de nodos amigos (aquellos que quieren mantener actualizados en sus *commit*).

## IV. Conclusiones

IPFS es un sistema para la permanencia de la información en la web, como primer objetivo. Mediante el uso de un sistema llamado IPFS Merkle DAG, el sistema logra dar identidad a los archivos mediante su contenido, y los ordena en un orden jerarquico, como si fuera un gran sistema de archivos global. Así, y con el uso de un sistema basado en Git se logra obtener un direccionamiento mediante contenido, y no por ubicación. A través de su *Peer Swarn* IPFS logra distribuir contenido, y mediante BitSwap logra hacer peticiones a *Peer Swarn*, de forma que no se accede a un único punto de la red para encontrar contenido, si no que se pide cual contenido se desea descargar. En definitiva, a través de IPFS se responde un que y no un donde, como ocurre con HTTP.

Es por ello que IPFS se ve como una buena alternativa para reemplazar a HTTP, cubriendo falencias como:

- Punto único de falla de acceso a la información.
- Uso innecesario de ancho de banda en mensajería de sincronización, e.g ICMP.
- Volatilidad de información. Un link roto puede dejar páginas web obsoletas.

Si se considera que cada día son mas los dispositivos con acceso a internet, va a ser necesario optimizar el uso de recursos en el *Backbone* de internet. El uso de un sistema descentralizado de información parece una buena aproximación. En la actualidad, IPFS cuenta con una *suite* completa para ser usado, incluyendo un cliente en terminal y acceso a archivos mediante navegador web.

El principal *trade-off* de este protocolo es que no se puede borrar información de la red, por lo que si alguna usuario comparte información accidentalmente, no puede arrepentirse de dicha acción. Otra desventaja es el acceso a la información en capa aplicación, pues al utilizar hash largos no es útil a la hora de compartir. Como ventaja, cuenta con la permanencia de la información en una red distribuida, y al eliminar el único punto de fallo, evita la autoridad sobre información, evitando problemas como censura y control de tráfico.

## Búsquedas Bibliográficas

[1] I. Baumgart y S. Mies., *S/kademlia: A practicable approach towards secure key-based routing*, en In Parallel and Distributed Systems, 2007 International Conference on, volume 2, págs 1-8. IEEE. 2007.

[2] BENET, JUAN, *IPFS - Content Addressed, Versioned, P2P File System*.
[3] HTTP://WWW.CORALCDN.ORG/.
[4] HTTPS://WWW.IETF.ORG/PROCEEDINGS/65/SLIDES/PLENARYT-2.PDF.
[5] R. R. STEWART AND Q. XIE, *Stream control transmission protocol (SCTP): a reference guide. Addison-Wesley Longman Publishing Co., Inc.,*, 2001.
[6] HTTPS://GIT-SCM.COM/.
[7] J. ROSENBERG AND A. KERANEN., *Interactive connectivity establishment (ice): A protocol for network address translator (nat) traversal for offer/answer protocols.*, 2013.
[8] HTTPS://TOOLS.IETF.ORG/HTML/RFC2104.
[9] HTTPS://BITCOIN.ORG/BITCOIN.PDF.