La web distribuida: el protocolo IPFS

Fundamentos de Redes

José María Martín Luque
Adolfo Soto Werner

18 de noviembre de 2017

Introducción

Introducción

HTTP es el protocolo de comunicación utilizado actualmente para la transferencia de información en la web.

El desarrollo de HTTP comenzó en 1989 en el CERN por parte de Tim Berners-Lee.

La primera definición de HTTP/1.1, la versión más utilizada apareció en 1997.

Introducción

Es un protocolo antiguo que no fue diseñado sin vistas al futuro y pensando Internet a menor escala y no tal y como lo vemos hoy en día.

Problemas de HTTP

Not Found

The requested URL /hola was not found on this server.

Apache/2.4.18 (Ubuntu) Server at dgiimcloud.ml Port 443

Fragilidad

HTTP es frágil porque para acceder al contenido se depende de una serie de servidores concretos.

Si alguno de ellos falla, no se puede acceder al contenido.

Fragilidad

Si un servidor cambia de dirección o deja de estar disponible, los enlaces que apuntan hacia él dejan de funcionar.

El creador de Pinboard estima que alrededor del 5 % de los enlaces que se almacenan en este servicio dejan de funcionar cada año.

Hipercentralización

La web actualmente está altamente centralizada. La práctica totalidad de los usuarios de Internet dependen de una serie de servicios concretos.

Una caída del servicio de Google en 2013 provocó una reducción mundial del tráfico de Internet del 40 %.

Hipercentralización

No hay que irse muy lejos.

Hipercentralización

La *hipercentralización* también facilita el control de las comunicaciones de los ciudadanos por parte de los Gobiernos.

Ineficiencia

¿Cuál sería el coste aproximado por haber distribuido el vídeo más visto de YouTube?

(Suponiendo un coste 1 céntimo/GB distribuido y que siempre se reproduce a 720p)

Ineficiencia

El vídeo en 720p pesa 61,1MB.

El vídeo se ha reproducido al menos 4.000.000.000 de veces.

$$((67,1 \cdot 4{,}000{,}000{,}000) \div 1024) \cdot 0,01 = 2.621.093,75€$$

La web distribuida

¿Qué es?

La idea de la web distribuida es eliminar la dependencia en una serie de servidores centrales haciendo que cualquier receptor sea a la vez emisor de información.

Un ejemplo en la naturaleza que ilustra esta idea es la de la ramificación de las neuronas.

Tecnologías de web distribuida

Blockchain: Ethereum

Comunicación: BitMessage

Bases de datos: IPDB

Web descentralizada: Tor

Difusión y almacenamiento de información: IPFS

...

El protocolo IPFS

¿Qué es?

IPFS es un sistema de archivos distribuido.

Los nodos IPFS se conectan entre sí para transferir datos.

El protocolo está dividido en un serie de sub-protocolos.

Identidades

Gestiona la generación y verificación de la identidad de los nodos.

Cada nodo se identifica por un `NodeID` que es el *hash* de su clave pública. IPFS no utiliza una función *hash* concreta.

Listing 1: Definición de Nodo.

```
1       // Hash criptográfico
2       type NodeId Multihash
3       type Multihash []byte
4
5       // Claves
6       type PublicKey []byte
7       type PrivateKey []byte
8
9       type Node struct {
10        NodeId NodeID
11        PubKey PublicKey
12        PriKey PrivateKey
13      }
```

Red

El susbsistema de red de IPFS incluye funciones para:

- Transporte
- Fiabilidad
- Conectividad
- Integridad
- Autenticidad

Red

IPFS puede utilizar cualquier red: no depende ni asume acceso a IP.

Enrutamiento

IPFS utiliza una Tabla Hash Distribuida (DSHT).

Listing 2: Interfaz de la DSHT.

```
type IPFSRouting interface {
  // Obtiene la dirección de un nodo concreto
  FindPeer(node NodeId)

  // Almacena un pequeño dato en la DSHT
  SetValue(key []byte, value []byte)

  // Obtiene un pequeño dato de la DSHT
  GetValue(key []byte)

  // Anuncia que el nodo puede distribuir un dato grande
  ProvideValue(key Multihash)

  // Obtiene una serie de peers distribuyendo un dato grande
  FindValuePeers(key Multihash, min int)
}
```

Intercambio de bloques: el protocolo BitSwap

BitSwap es el protocolo de intercambio de bloques de IPFS. Está inspirado en BitTorrent.

Los *peers* buscan conseguir un conjunto de bloques (`want_list`) y tienen otro conjunto de bloques que ofrecer (`have_list`).

BitSwap actúa como una especie de *mercado*.

Normalmente el intercambio de archivos no es complementario.

Crédito BitSwap

Los nodos envían bloques a sus *peers* de forma optimista.

Hay que protegerse de los *leeches*.

Un sistema de créditos resuelve el problema:

- Los *peers* realizan un seguimiento de su saldo con otros nodos.
- Los *peers* envían bloques a otros *peers* probabilísticamente.

Si un nodo decide no enviar bloques a un *peer* el nodo lo ignorará durante un tiempo.

Estrategia BitSwap

La elección de una función debe procurar varias cosas.

Una función que es efectiva en la práctica es una sigmoide, escalada por una *ratio de deuda*.

$$r = \frac{\texttt{bytes\_enviados}}{\texttt{bytes\_recibidos} + 1}$$

Dado r la probabilidad de enviar a un deudor es:

$$P(r) = 1 - \frac{1}{1 + exp(6 - 3r)}$$

Estrategia BitSwap

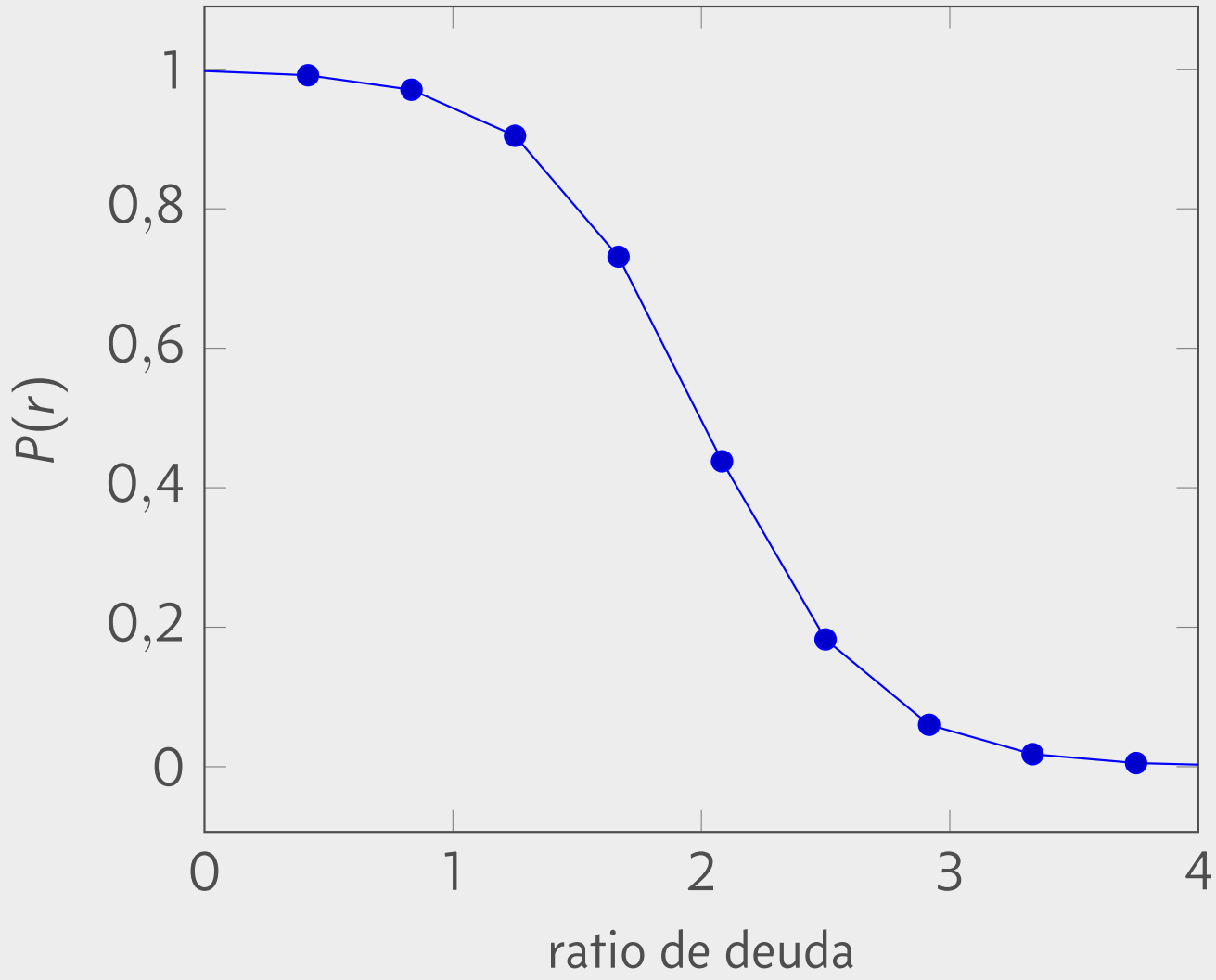

Figura 1: Representación de la función de probabilidad $P(r)$.

Libro mayor de BitSwap

Los nodos mantienen un registro de las transferencias con otros nodos.

Cuando se inicia una conexión entre dos nodos intercambian su libro mayor.

Si no concuerdan, el libro mayor se destruye. Si la pérdida es intencionada puede que los nodos decidan negar el intercambio de información.

Listing 3: Implementación del libro mayor.

```
type Ledger struct {
  owner NodeId
  partner NodeId
  bytes_sent int
  bytes_recv int
  timestamp Timestamp
}
```

Conexión BitSwap

Esquema de conexión con un *peer*:

1. Conexión: los *peers* envían sus libros mayores hasta que estén de acuerdo.
2. Envío: los *peers* intercambian sus listas y los bloques.
3. Cierre: los *peers* desactivan la conexión.
4. Ignorado: un *peer* es ignorado si el nodo decide no enviar información.

Grafo Dirigido Acíclico de Merkle

La implementación del sistema de archivos de IPFS es una generalización de la estructura de datos de Git.

Utilizar un GAD de Merkle nos aporta:

- Direccionamiento de contenido
- Resistencia a la manipulación
- Unicidad

Rutas

Los objetos IPFS pueden ser recorridos con una API de rutas *string* con el siguiente formato:

Listing 4: Rutas en IPFS.

```
// Formato
/ipfs/<hash-del-objeto>/<nombre-del-objeto>

// Ejemplo
/ipfs/XLYkgq61DYaQ8NhkcqyU7rLcnSa7dSHQ16x/foo.txt
```

Archivos: `blob`

El objeto `blob` contiene una unidad de datos direccionable

Representa un archivo sencillo.

Listing 5: Estructura JSON de un `blob`.

```
1    {
2      "data": "datos aquí",
3    }
```

Archivos: `list`

El objeto `list` representa un archivo grande, formado a partir de varios `blobs`.

A diferencia de `blob`, `list` tiene enlaces.

Listing 6: Estructura JSON de un `list`.

```
{
  // Las listas pueden tener un array de objetos como datos
  "data": ["blob", "list", "blob"],

  // Las listas no tienen nombres en los enlaces
  "links": [
    { "hash": "nccSbCKcEWHAJ9wjOM72", "size": 38231 },
    { "hash": "2dUB2OS5O7rJBI57n7Hs", "size": 110 },
    { "hash": "BHHGhOpUmwiAyjN5mbTY", "size": 6468 },
  ]
}
```

Archivos: `tree`

El objeto `tree` representa un directorio. Es un *mapeo* de nombres a *hashes*.

Listing 7: Estructura JSON de un `tree`.

```
{
  // Los árboles pueden tener un array de objetos como datos
  "data": ["blob", "list", "blob"],

  // Las listas no tienen nombres en los enlaces
  "links": [
    { "hash": "nccSbCKcEWHAJ9wj0M72", "name": "test1", "size": 38231 },
    { "hash": "2dUB2OS5O7rJBI57n7Hs", "name": "test2", "size": 110 },
    { "hash": "BHHGh0pUmwiAyjN5mbTY", "name": "test3", "size": 6468 },
  ]
}
```

Archivos: `commit`

El objeto `tree` representa una instantánea en el historial de versiones de cualquier objeto.

Puede hacer referencia a cualquier tipo de objeto.

Listing 8: Estructura JSON de un `commit`.

```
{
  "data": {
    "type": "tree",
    "date": "2017-11-12 19:58:00",
    "message": "El mensaje del commit."
  },
  "links": [
    { "hash": "nccSbCKcEWHAJ9wj0M72", "name": "parent", "size": 38231 },
    { "hash": "2dUB2OS5O7rJBI57n7Hs", "name": "object", "size": 110 },
    { "hash": "BHHGh0pUmwiAyjN5mbTY", "name": "author", "size": 6468 },
  ]
}
```

La web distribuida en la actualidad