Estudio de BITTORRENT

Autores

Oscar Barrios

Jonathan Hernández

Que es BITTORRENT?

- Protocolo de distribución de ficheros P2P
- Autor: Bram Cohen
- Programa cliente oficial
  - Open Source
  - Escrito en Python

Que lo diferencia?

- Optimizado para la replicación y distribución de contenidos
- Mayor velocidad de descarga
- No implementa búsqueda de contenidos
- Resulta difícil encontrar material que no sea novedoso

Incentivos para compartir

- Sigue una estrategia “Tit for Tat”
- Negociación entre peers
- Si un peer no comparte su información, los peer con los que mantiene comunicación actuaran del mismo modo
- Utiliza una variante que perdona al peer egoísta cada N segundos, si este ha comenzado a compartir

Arquitectura

Web Server

Tracker

Peer [Leech]

Peer [Seed]

Peer [Leech]

Arquitectura

Web Server

Tracker

Cliente que aún no tiene el fichero completo

Peer [Leech]

Peer [Seed]

Peer [Leech]

Arquitectura

Web Server

Tracker

Cliente que
comparte todas las
partes de un fichero

Peer [Leech]

Peer [Seed]

Peer [Leech]

Arquitectura

Web Server

Tracker

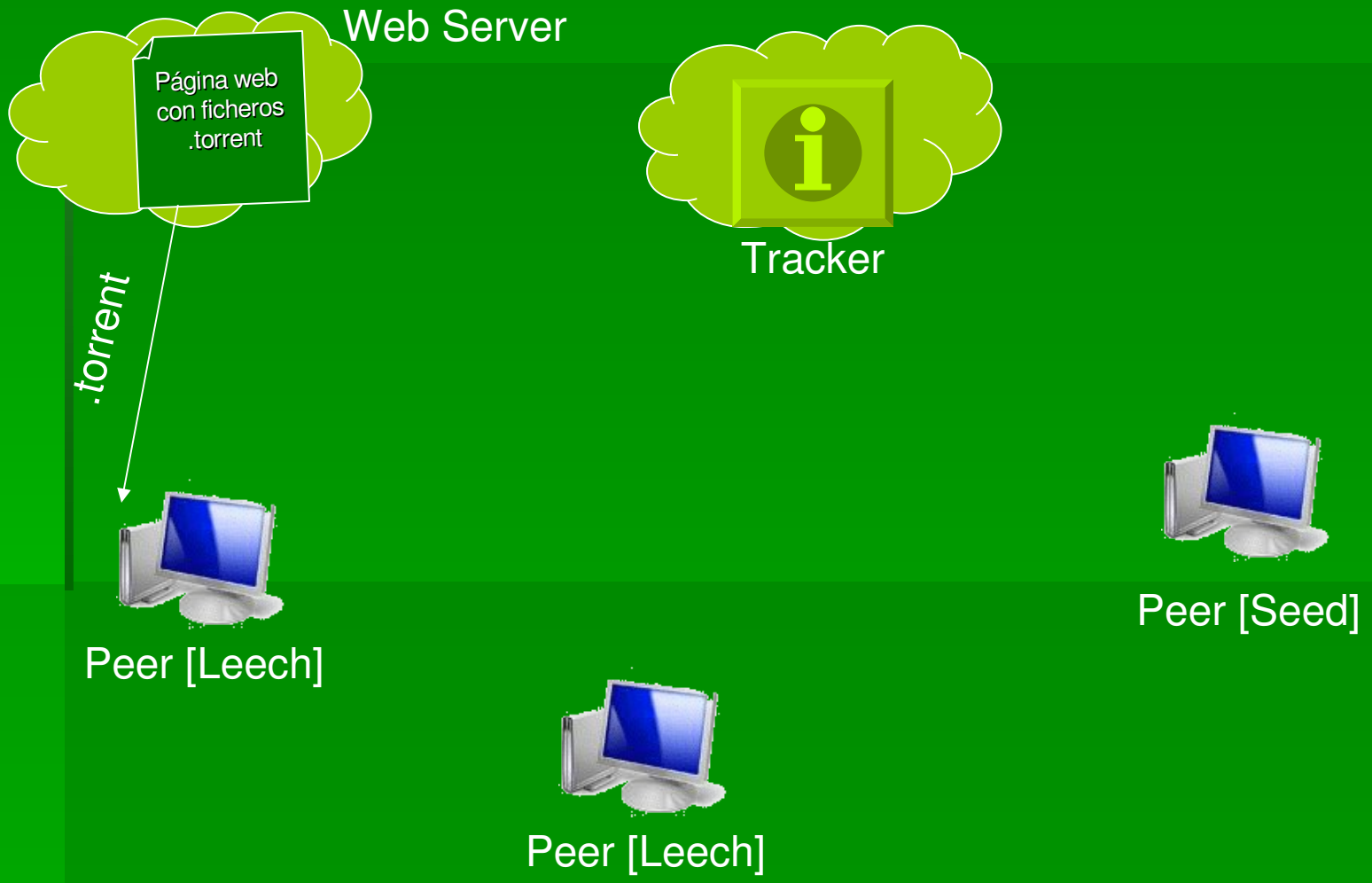
Único punto de coordinación entre peers que comparten un mismo fichero

Peer [Leech]

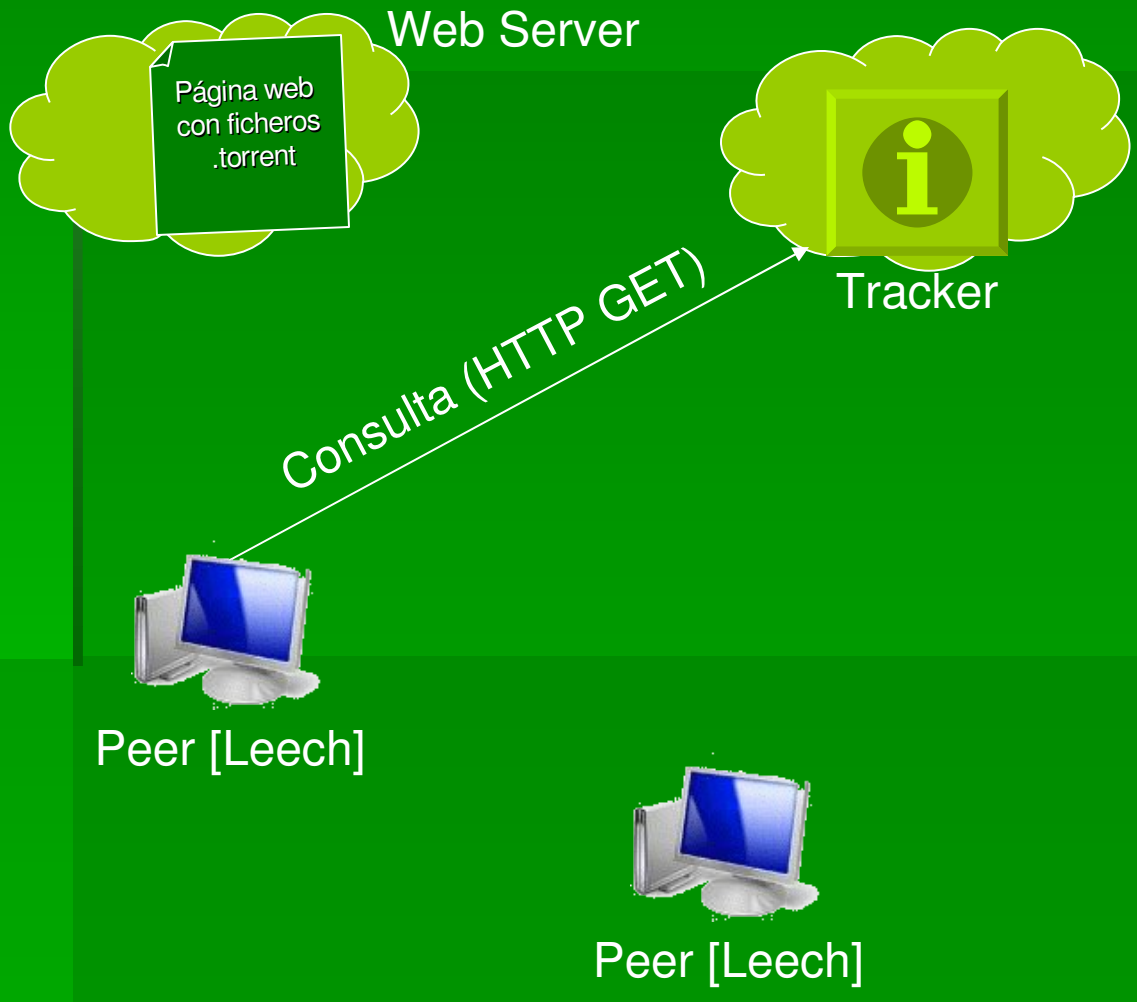
Peer [Seed]

Peer [Leech]

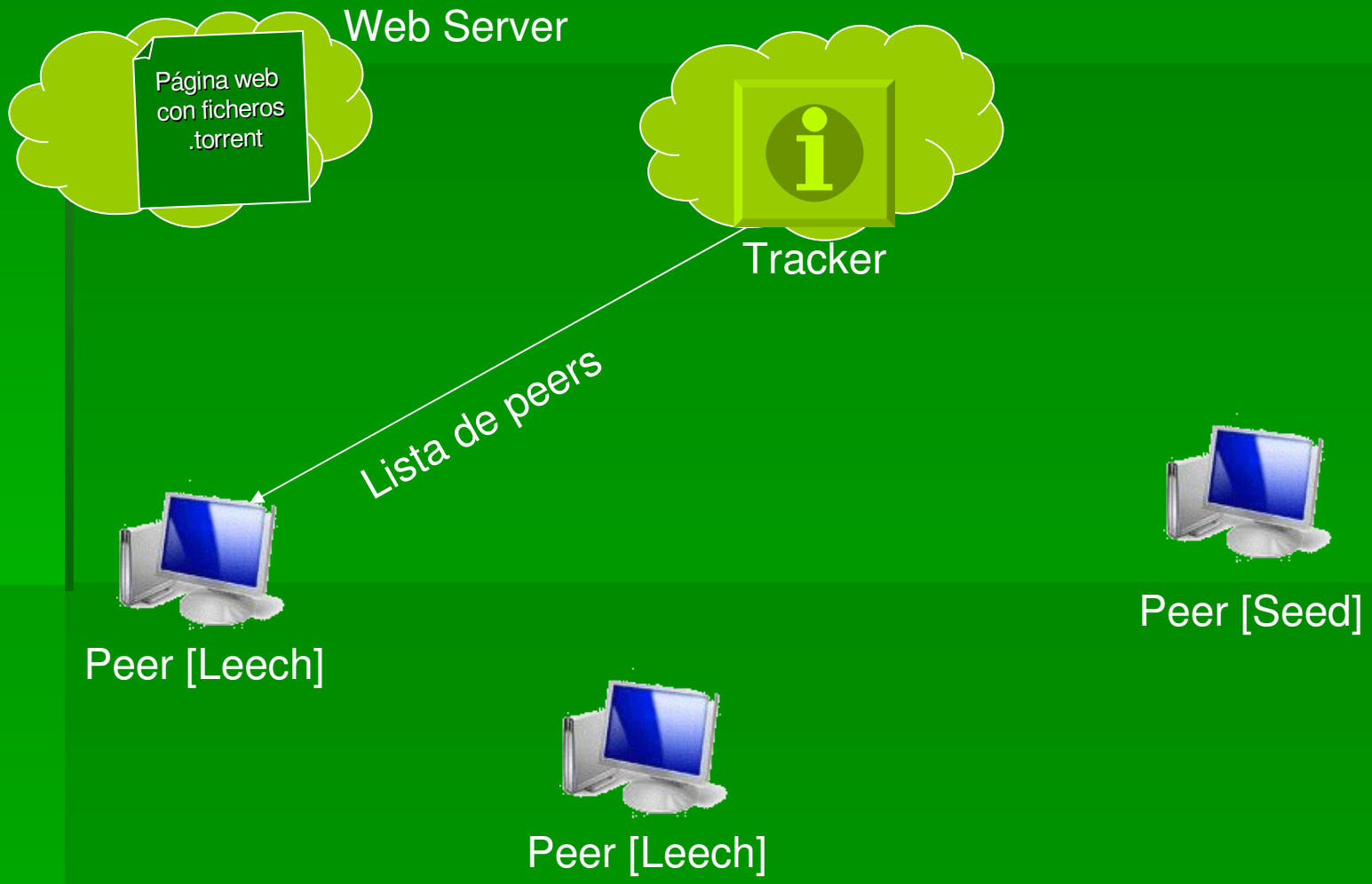
Arquitectura



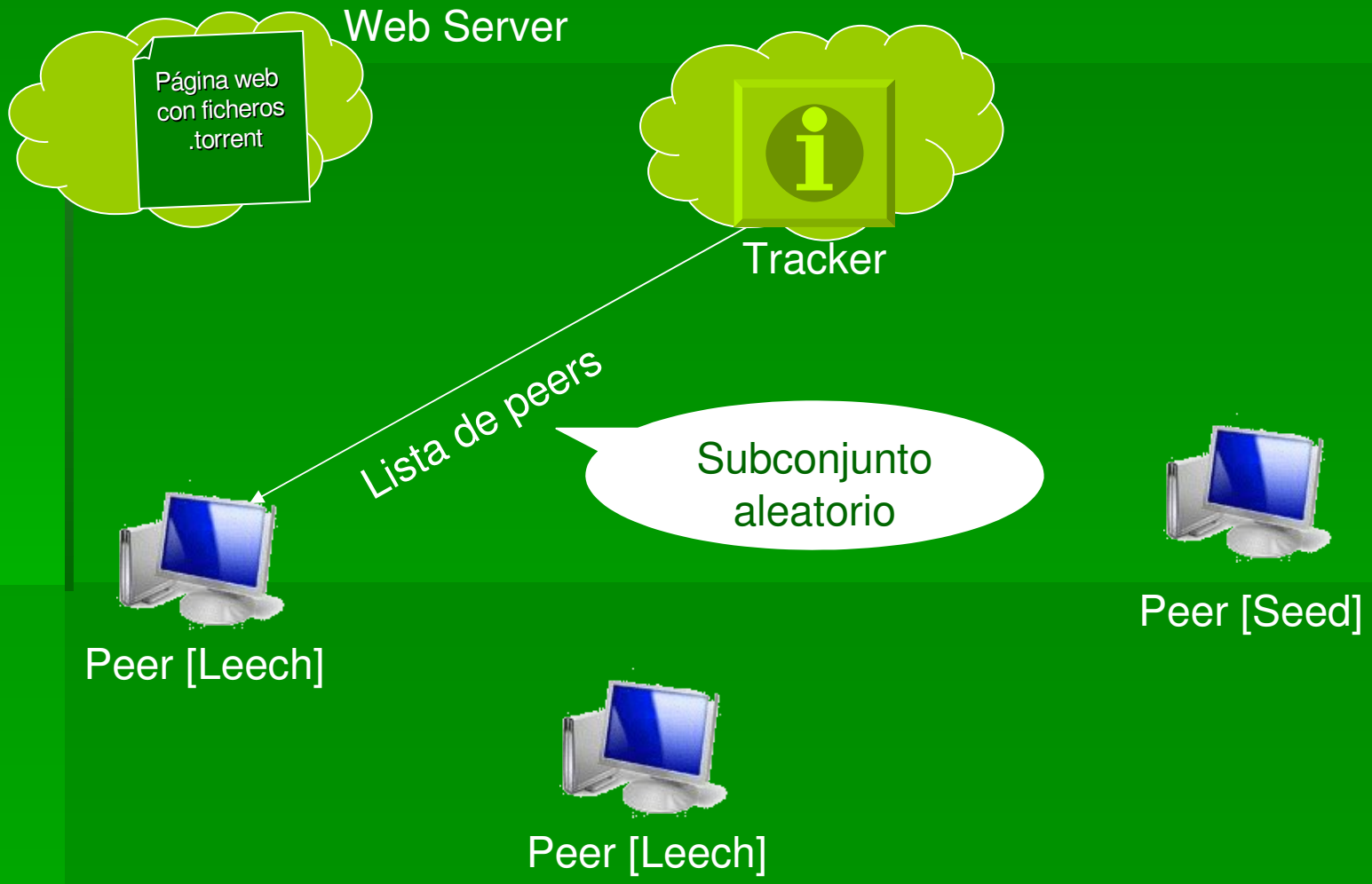
Arquitectura



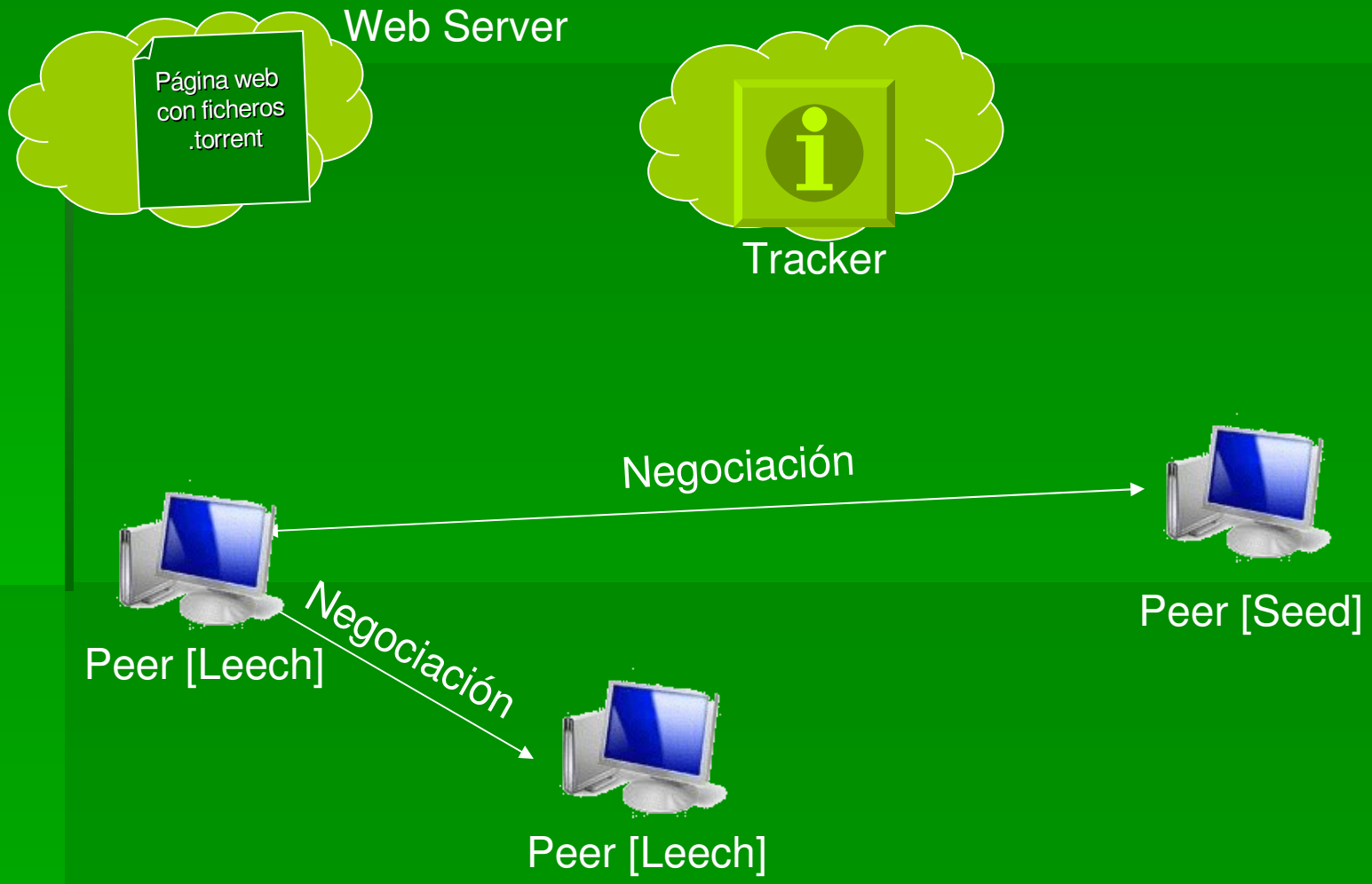
Arquitectura



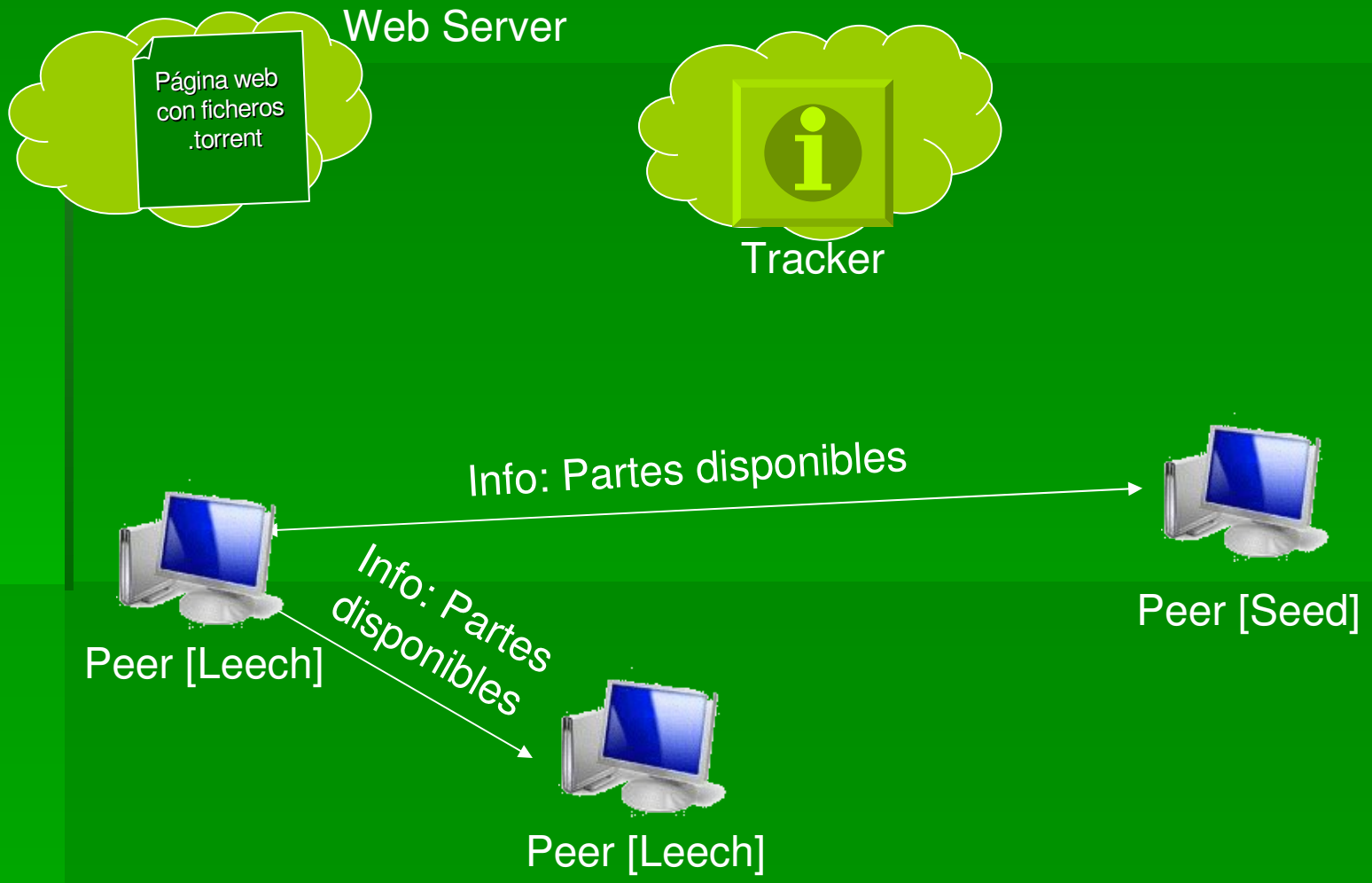
Arquitectura



Arquitectura

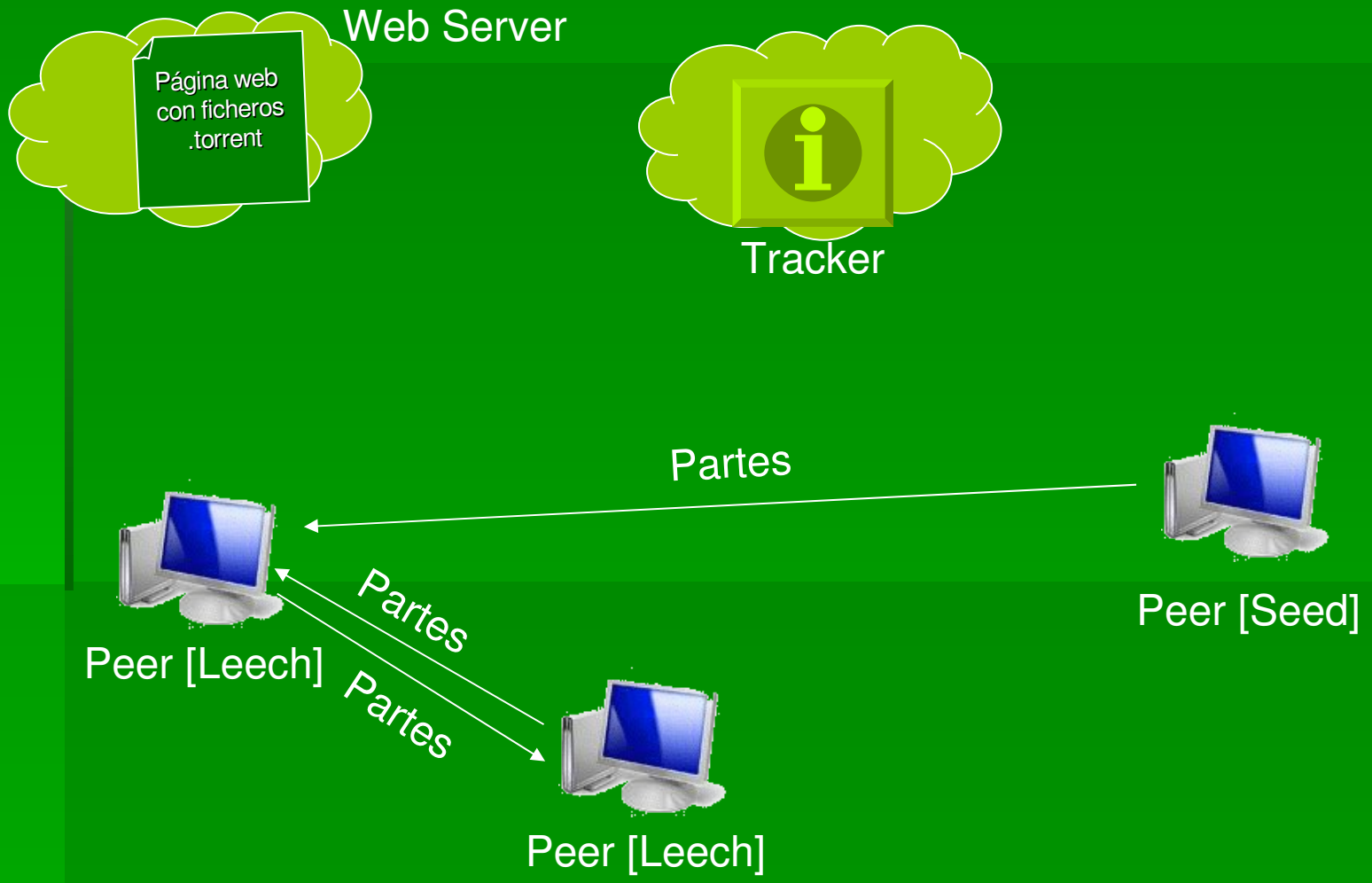


Arquitectura

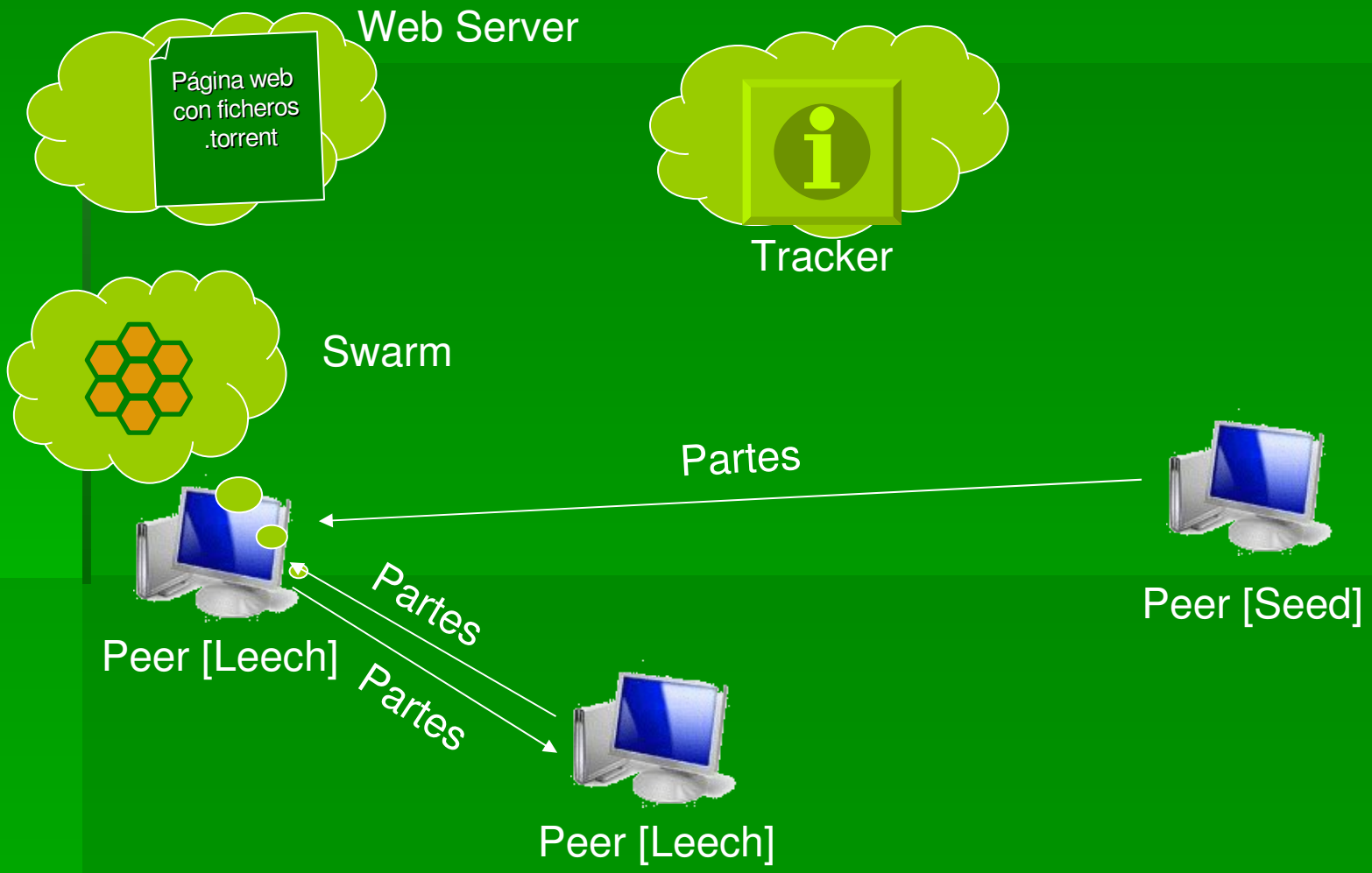


Arquitectura

Arquitectura



Arquitectura



Comunicación

- Peer – Peer mensajes
  - TCP Sockets
- Peer – Tracker mensajes
  - HTTP Petición/Respuesta
- B-encoding:
  - Codificación propia para estructurar metainformación, soporta :
    - Cadena de bytes, enteros, listas y diccionarios

Swarming

- Todos los peers comparten aquellas partes del fichero que poseen, de forma inmediata a la descarga
- Comparten partes simultáneamente entre varios peers
- Técnica utilizada por otras redes como eDonkey

Ficheros .torrent

- URL del tracker
- Lista de claves hash por cada parte del contenido
- Tamaño de una parte
- Nombre
- Tamaño del contenido
- Fichero/s que contiene
  - Nombre (puede incluir un path)
  - Tamaño en bytes

Ficheros .torrent

- Las claves hash son generadas utilizando SHA-1 :
  - Resumen de : 160 bits
  - Tamaño máximo de cada parte : 2^{64} bits
- Utiliza las claves como mecanismo para asegurar la consistencia de datos en la descarga de cada parte

Tipos de Mensaje

- En la comunicación entre peers, se establece un conjunto de identificadores de mensaje
- Permiten compartir las partes de un fichero
- Además, ofrecen soporte para seguir una estrategia Tit for Tat

Tipos de Mensaje

- **Keep-alive:**
  - Mensaje de longitud cero, enviado para mantener la comunicación
- **Choke / Unchocke:**
  - Mensaje que comunica al receptor de un bloqueo temporal en la transferencia de información
- **Interested / Not Interested:**
  - Mensaje que comunica de un interés por parte del emisor en los ficheros que el receptor ofrece

Tipos de Mensaje

- **Have:**
  - Mensaje enviado cuando una parte esta descargada y verificada, y por tanto puede compartirse
- **Bitfield:**
  - Solamente puede enviarse inmediatamente después de establecer la conexión. Envía la lista de partes que puede compartir, en caso de tener alguna
- **Request:**
  - Mensaje de petición de un bloque de datos, se especifica el tamaño del bloque, el número de parte y un offset

Tipos de Mensaje

- **Piece:**
  - Mensaje que envía un bloque de datos de una parte, se especifica el número de pieza, un offset y la medida del bloque
- **Cancel:**
  - Mensaje para cancelar las peticiones de bloques
- **Port:**
  - Mensaje que informa del puerto de escucha si este peer se encuentra como parte de un DHT tracker (soportado por las últimas versiones)

DHT y BITTORRENT

- **Problema del cuello de botella**
  - ¡¡Todos contra el tracker!!
- **Concepto de DHT Tracker:**
  - Distribuir la carga del tracker entre los peers. Capa independiente del protocolo Bittorrent.
  - Antes sólo había caché individual de peers desde el cliente para responder ante fallo del tracker.

DHT y BITTORRENT

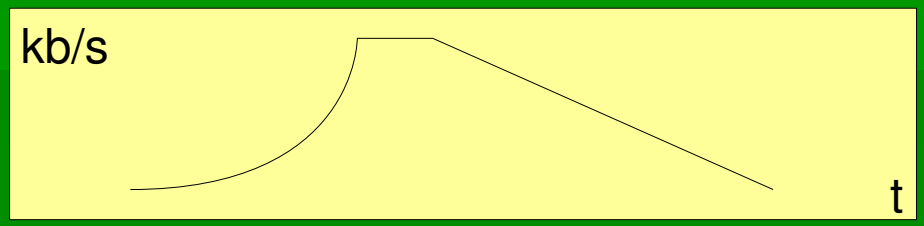
- **Características**
  - Utiliza un puerto UDP para la comunicación entre nodos.
  - El ID del nodo es el algoritmo de SHA-1 aplicado a la IP y el puerto utilizado de cada nodo.
- **Implementaciones:**
  - Azureus, Bittorrent 4.1.2 beta+, Bitcomet ...etc.
    - Como base Kademlia, pero extendido.
    - Incompatibles... why??

DHT y BITTORRENT

- **¿Modelo descentralizado?**
  - Tracker y .torrent necesarios para entrar
    - Magnet links en Azureus
- **Problemas con los trackers privados**
  - Control del contenido entre usuarios/IP registradas.
  - Private flag
- **Otras utilidades**
  - Ratings y Comments

Comportamiento

- **Escalabilidad**
  - Con millones de peers, el **tracker** utiliza 1/1000 de ancho de banda. No es poco.
  - Miles de peers desde el principio sigue una distribución de este tipo:



  - La escalabilidad depende mayoritariamente del ancho de banda del Tracker. DHT ayuda.

Comportamiento

- **Robustez y tolerancia a fallidas**
  - Si el tracker cae:
    - Nuevos nodos no pueden conectarse
    - Nodos conectados no descubren nuevos nodos
      - Pequeñas islas de nodos

Comportamiento

- **Ataques sobre Bittorrent**
  - Modelo P2P económico – Es inocente
    - Soluciones a nivel de cliente
  - Malicious upload attack
    - Envio constante de archivos corruptos.
    - Vigiliar a nivel de cliente y bloquear la IP.
  - Sybil attack
    - ID Bittorent = Hash de IP+Time
    - Múltiples identidades desde un cliente - DoS

Comportamiento

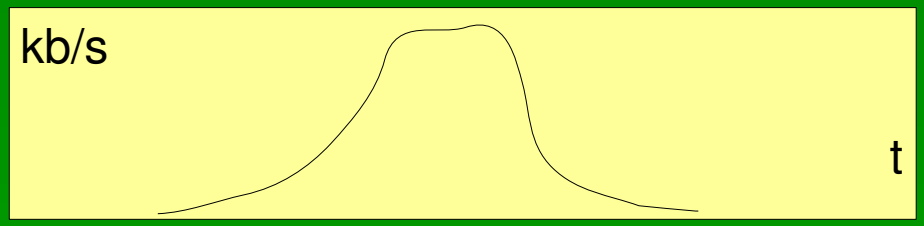
- **Ataques sobre Bittorrent**
  - Seed only attack
    - Se intenta conectar sólo a seeds. Bittfield a 1's indica que es un seed. Fáciles de identificar.
    - Sin necesidad de subir nada.
  - Minimal upload attack
    - Mismo incentivo si subes a 1kb/s que a 10kb/s
    - Objetivo, mínimo para entrar en la lista de preferred peers de los otros nodos.
    - Combinado con el Sybil attack.

Limitaciones

- **Confía en la gente**
  - **Incentivos sólo para los peers**
    - No hay motivación para los seeds, altruísmo.
    - 20%-40% usuarios Napster comparten poco o nada. 70% en el caso de Gnutella.
    - Desconocimiento.
  - **Nettiquette**
    - Esta mal visto dejar de compartir un recurso cuando has acabado de descargarlo si el índice de compartición < 1.0

Limitaciones

- **¿Afecta al rendimiento?**
  - Oh God, yes.



- **Private trackers**
  - Restrictivos respecto al índice de compartición

Limitaciones

- **Descarga partes al azar**
  - Por defecto sigue un algoritmo en que tienen prioridad los “trozos más raros” (los que tienen menos peers).
  - También el modo SuperCompartición, donde se priorizan las partes que no han sido todavía descargadas (al principio).
    - Buenos para distribuir rápidamente el recurso
  - Imposibilita el progressive download o el streaming playback. Solución: priorizar primeras partes.

Limitaciones

- **Boot-strapping**
  - Sigue algoritmo TIT for TAT, so...
    - Pag 4. "*Si un peer no comparte su información, los peer con los que mantiene comunicación actuaran del mismo modo*"
    - Nuevos nodos no tienen aún información...

Una solución quiero

Limitaciones

- **Optimistic unchoking (i)**
  - Los peers tienen su lista de N nodos más rápidos de los que descarga/sube información: **Preferred peer list**.
  - Pero también son curiosos, periódicamente intentan descubrir nodos más rápidos. % Upload destinado.

Limitaciones

- **Optimistic unchoking (ii)**
  - Un nuevo nodo tiene que esperar a que los demás por “optimistic unchoke” le complete almenos una parte para poder empezar a compartir y formar parte del TIT for TAT.
  - Suele tardar entre 5-10 minutos en estabilizarse la lista de preferred peers.

The End

¿Preguntas?