
Sistemas Distribuidos

Práctica en grupo

BitCascade

Objetivo

- Práctica en pareja con nota hasta 12 puntos
- Versión simplificada de BitTorrent con Java/RMI
 - Descarga de fichero de gran tamaño usando esquema P2P
- Arquitectura con 3 tipos de nodos:
 - Publisher/Seed: publica un fichero
 - Downloader/Leech: descarga un fichero
 - Tracker: mantiene metainformación de fichero descargado
- Modo de operación:
 - Publisher contacta con tracker enviándole metainformación del fichero
 - Incluye los datos de contacto del publisher
 - Downloader obtiene metainformación del fichero
 - Incluye los datos de contacto del publisher y de los downloaders previos
 - Downloader descarga los bloques del fichero
 - Tanto del publisher como de los downloaders previos

Organización del software

- Directorio *common* (paquete *interfaces*):
 - Incluye todas las interfaces del sistema que ya están programadas
 - Tracker.java: la interfaz del tracker.
 - Seed.java: la interfaz del publisher.
 - Leech.java: la interfaz de un downloader.
 - FileInfo.java: metainformación de un fichero.
 - Compiladas en common.jar
- Directorio tracker_node (paquete tracker):
 - funcionalidad del tracker: TrackerSrv.java.
- Directorio peer_node (paquete peers): contiene tres clases:
 - Publisher.java: la implementación del publisher.
 - DownloaderImpl.java: la implementación del downloader.
 - Downloader.java: usa servicios DownloaderImpl para descargar fichero.
- Solo hay que desarrollar las tres clases subrayadas

Componentes

interfaz remota

announce()
lookupFile()
addLeech()

tracker

registry

downloader

copia F

interfaz remota

read()
lastBlock()
notifyBlock()
newLeech()

leech

downloader

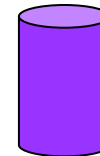
copia F

interfaz remota

read()
lastBlock()
notifyBlock()
newLeech()

leech

publisher



F

interfaz remota

read()

seed

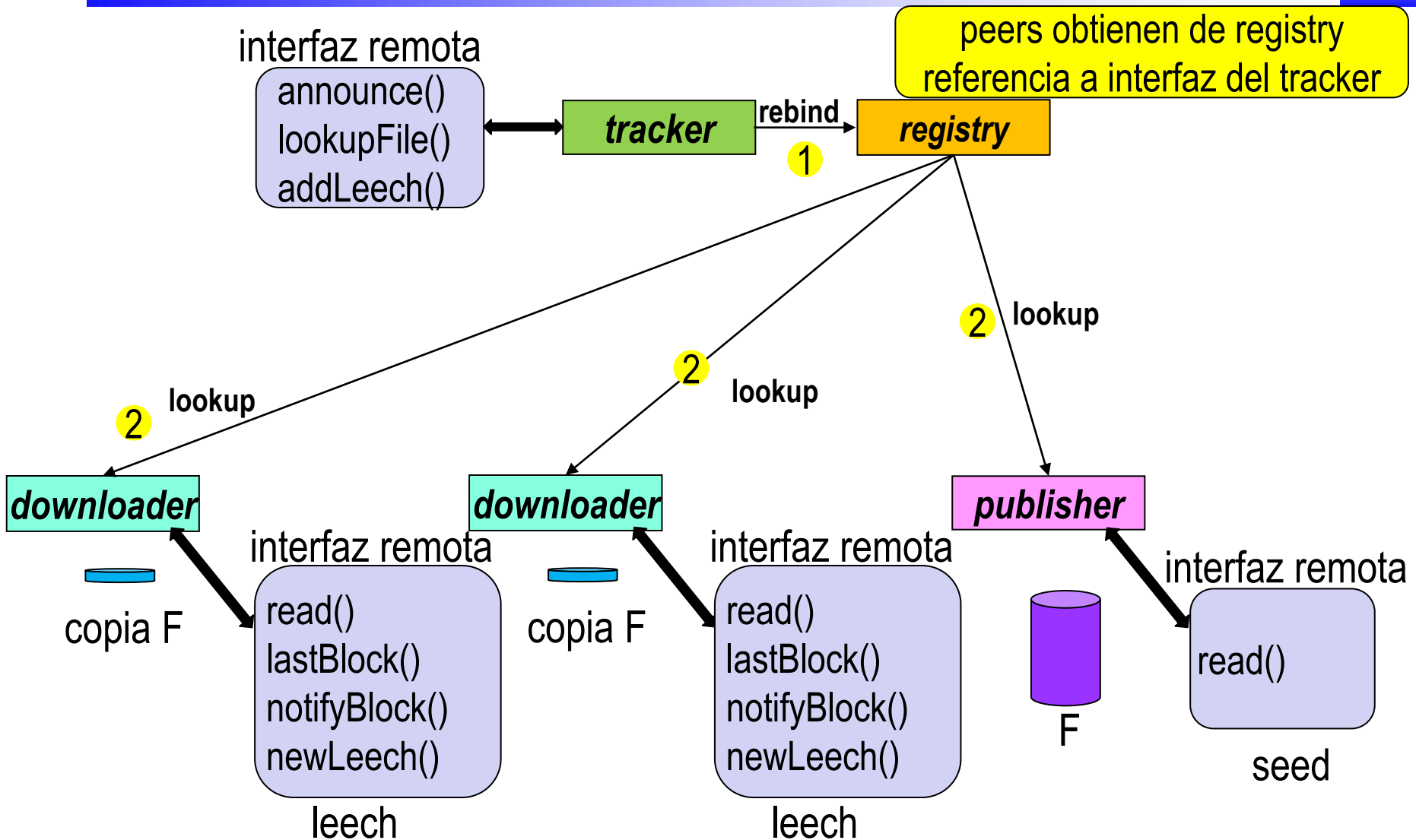
Ejecución de pruebas

```
wget https://laurel.datsi.fi.upm.es/~ssoo/SD.dir/practicas/BitCascade.tgz
tar xvf BitCascade.tgz; cd DATSI/SD/BitCascade.2023; ./compile_all.sh
cd tracker_node # en una primera ventana
./start_rmiregistry 23456 &
./execute_tracker.sh 23456 mi_tracker # nombre del tracker para depurar
cd peer_node # en una segunda ventana
mkdir bin/mi_publisher
cp /etc/passwd bin/mi_publisher/Fichero # fichero publicado
./execute_publisher.sh localhost 23456 mi_publisher Fichero 512
cd peer_node # en una tercera ventana
./execute_downloader.sh localhost 23456 down1 Fichero
# en peer_node/bin/down1/Fichero debe quedar una copia del fichero
# se lanzarían más downloaders en otras ventanas
```

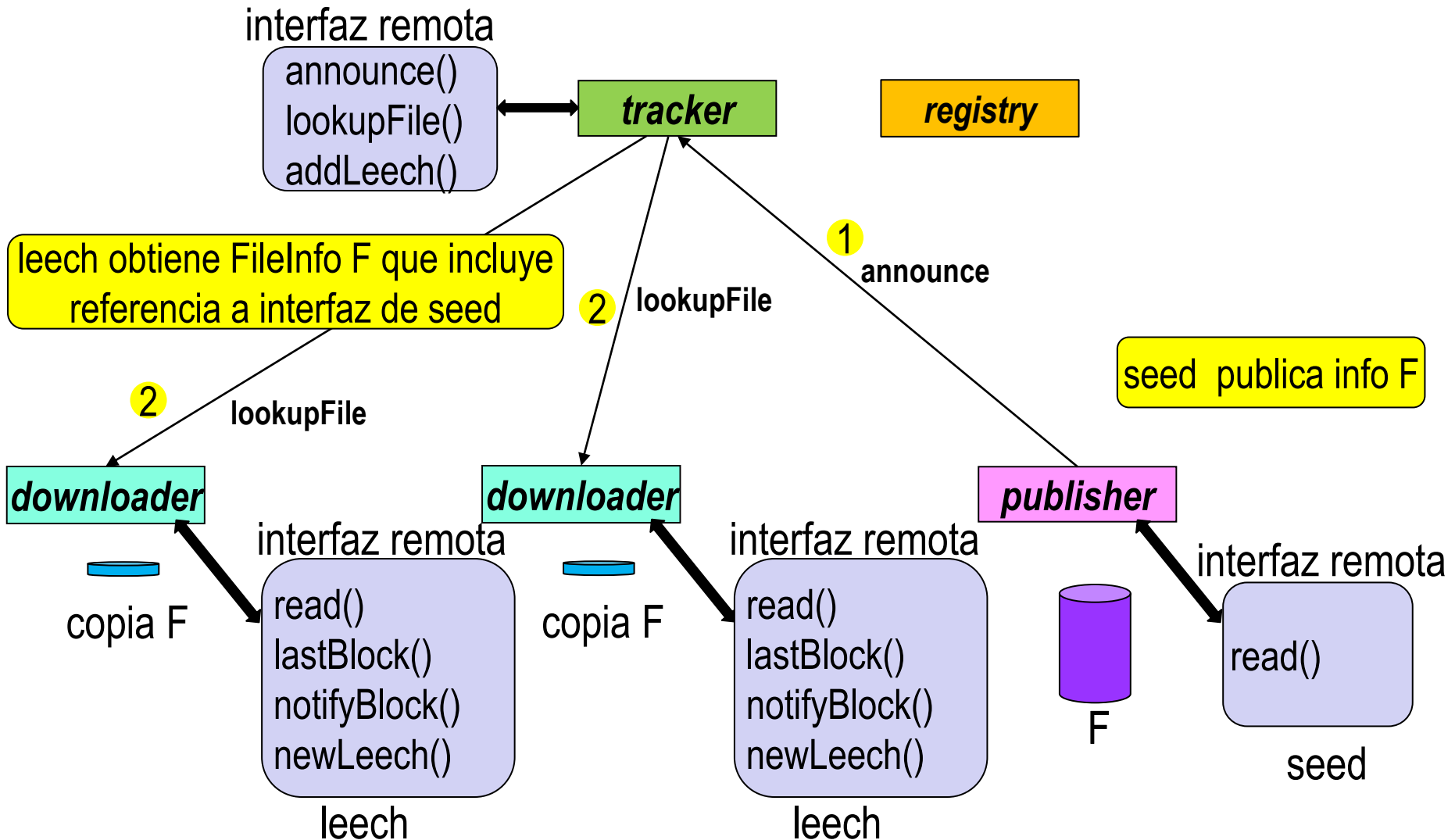
Fases de la práctica

- Fase 1 (3 puntos): Interacción con el tracker
- Fase 2 (3 puntos): Descarga de un fichero del seed
- Fase 3 (3 puntos): Descarga de leeches y de seed
 - Información estática
- Fase 4 (3 puntos): Descarga de leeches y de seed
 - Información dinámica

Fase 1: Operaciones en el *registry*



Fase 1: Operaciones en el *tracker*



Fase 2: Descarga solo desde seed

interfaz remota

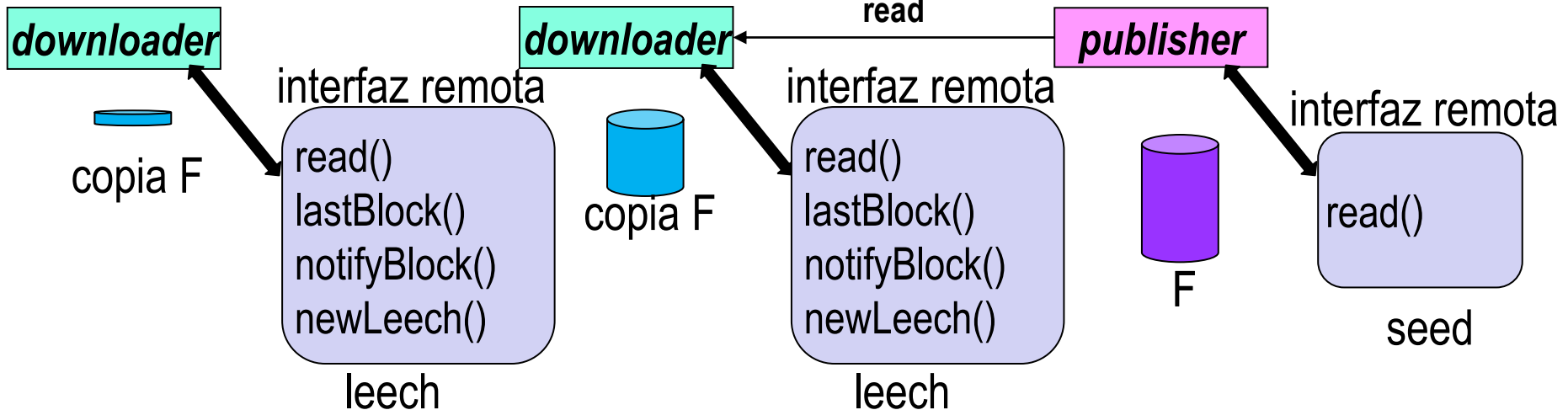
announce()
lookupFile()
addLeech()

tracker

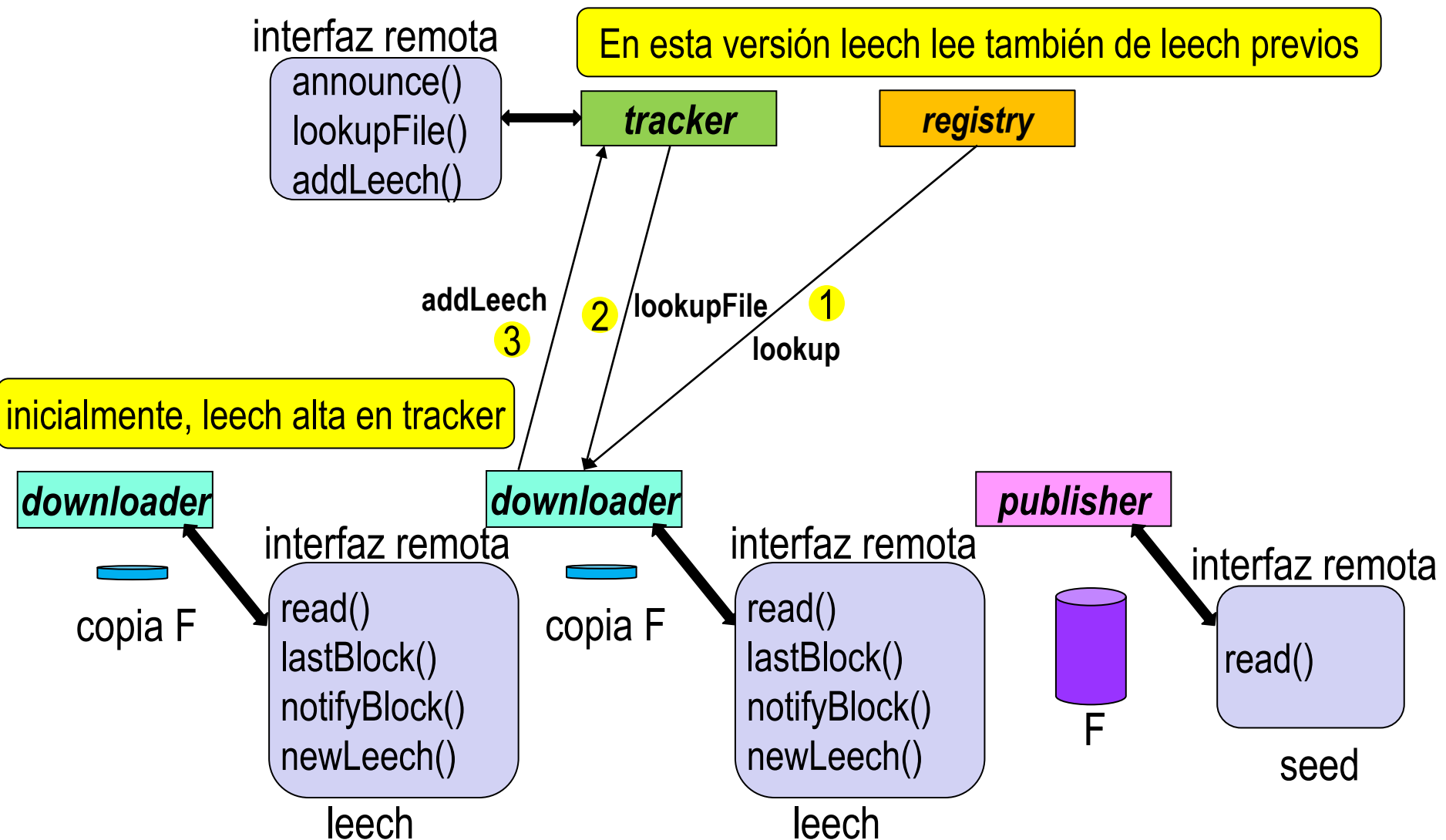
registry

En esta versión leech lee solo de seed

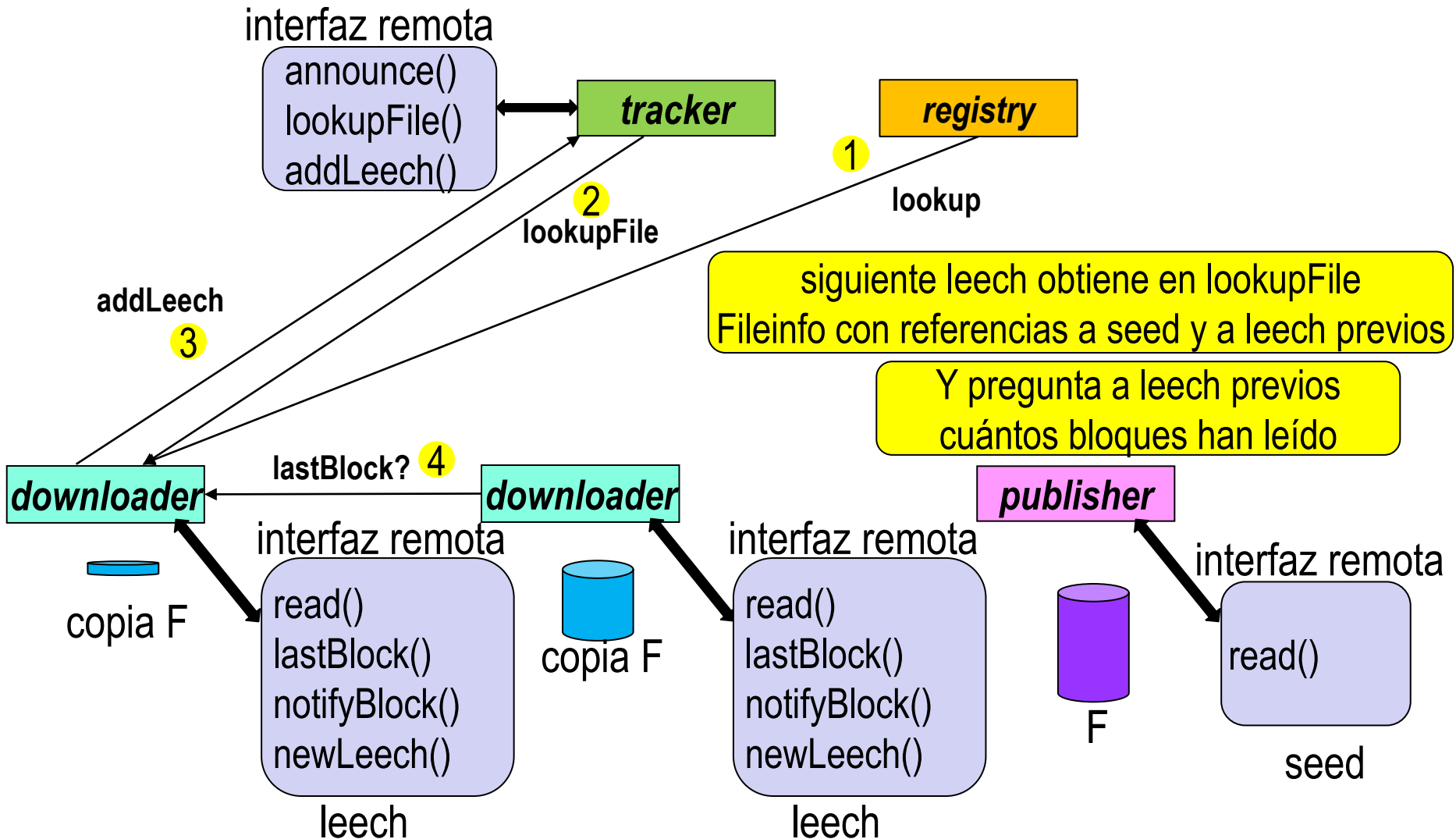
leech usa referencia a interfaz de seed
para descargarse los bloques



Fase 3: Descarga desde leeches y seed



Fase 3: Descarga desde leeches y seed



Fase 3: Descarga desde leeches y seed

interfaz remota

announce()
lookupFile()
addLeech()

tracker

registry

leech lee bloques de seed y de leech previos
pero solo sabe qué bloques tenían
cuando este leech se inició

read

read

downloader

downloader

publisher

copia F

interfaz remota

read()
lastBlock()
notifyBlock()
newLeech()

leech

copia F

interfaz remota

read()
lastBlock()
notifyBlock()
newLeech()

leech

F

interfaz remota

read()

seed

Fase 4: Notificaciones de los leeches

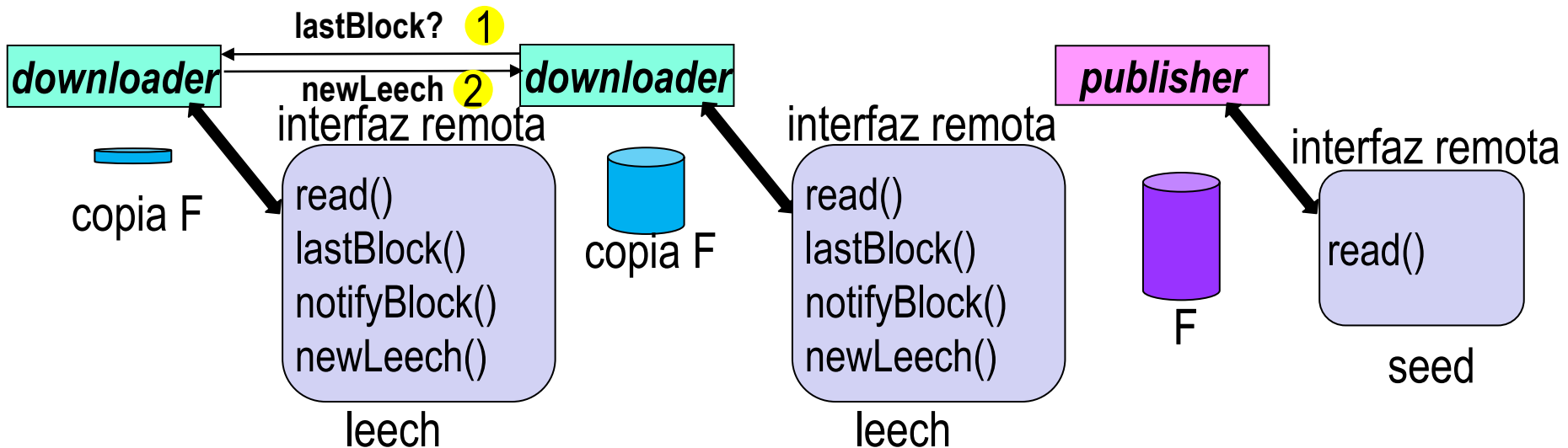
interfaz remota

announce()
lookupFile()
addLeech()

tracker

registry

leech pide ser notificado (addLeech)
cada vez que leech anterior descarga un bloque



Fase 4: Notificaciones de los leeches

interfaz remota

announce()
lookupFile()
addLeech()

tracker

registry

cada vez que un leech termina de leer un bloque
se lo notifica a leeches posteriores

