

Una vez el peer dispone de la información de las semillas para un fichero, puede solicitar a otros peers la descarga de trozos (de tamaño fijo) del fichero, a los que denominaremos normalmente como chunks, a través de un canal confiable. Utilizando multiformato o lenguaje de marcas especificar, al menos, los siguientes aspectos del protocolo peer to peer:

3. Puesto los peers pueden servir trozos de un fichero aunque no lo hayan descargado por completo, es necesario que un peer averigüe qué trozos pueden obtener de otro peer. Diseña los mensajes para dicha consulta y la correspondiente respuesta. Diseña los mensajes necesarios para solicitar y servir los trozos del fichero.

Formato del mensaje: SEEDQUERY para el tipo QUERY_CHUNKS

- Type = 7 (QUERY_CHUNKS)
 - Formato del mensaje: SEEDQUERY
 - Un peer solicita al otro peer la lista de chunks que tiene de un determinado fichero

Type (1 byte)	Size (5 bytes)	Filename length (2 bytes)
Filename (filename length bytes)		

Información del paquete:

- Type: Siempre sera 7 para indicar que es un QUERY_CHUNKS
- Size: Todo a 0 ya que no se usa
- Filename length: Longitud del nombre del fichero
- Filename (filename length bytes): nombre del fichero del que deseamos saber los chunks disponibles, puede que en vez del nombre del fichero podamos poner el Hash del fichero ya que evitaría fallos por nombre duplicado.

Formato del mensaje: CHUNKSRESPONSE para el tipo QUERY_CHUNKS_RESPONSE

- Type = 8 (QUERY_CHUNKS_RESPONSE)
 - Formato del mensaje: CHUNKSRESPONSE
 - Un peer informa a otro peer de la lista de chunks que tiene de un determinado fichero listo para compartir.

Type (1 byte)	Num Chunks (2 bytes)
Hash (20 bytes)	
Chunk (X bytes)	

Información del paquete:

- Type: Siempre sera 8 para indicar que es un QUERY_CHUNKS_RESPONSE
- Num Chunks: Numero de chunks que tiene para compartir
- Hash: Hash del fichero del que nos esta informando
- Chunk: numero del chunk que tiene disponible, se repetirá n veces, siendo n: Num Chunks