(a) Which option you have implemented.

I have implemented option 2.

(b) Clearly indicate whether you have tested your programs on a lab machine.

I have tested my programs on a lab machine, and it was working properly.

(c) Instructions on how to run your programs.

To run my programs: 1) run start_all.bat to open three WPF applications; 2) in the server application, click "Start Server"; 3) in the cache application, click "Start Cache Server"; 4) in the client application, click "Connect Server"; 5) upload some image files to start testing.

(d) Describe the techniques that you use to determine which portions of a file need to be downloaded from the server.

1. When client user is trying to download an image, client will send a request to cache.
2. When cache receive a request about an image file, it will send a request to server, asking for a list of fragments that can be used to re-construct the image file. Cache will always send such requests when it receive requests about image files from client, because server is the one who manages files and knows what fragments to use to construct an image. This design can make sure that even the content of a file with a same name has changed, cache can still know what fragments can be reuse by comparing the list it gets from the server and the fragments it has stored.
3. When server receive a request about a fragment list. It will find the list file based on file name and send it to cache.
4. When cache receive the list, it will compare the list with the list of fragments it has stored to find out what fragments are needed but missing. Then cache will send request to server asking for fragments one by one.
5. When server receive a request about a fragment, it will send the fragment to cache.
6. When cache receive a fragment, it will store the fragment. When cache gets all the fragment it needs to re-construct the image file that client was asking for, it will construct the image file and send it back to client.

(e) Consider the techniques that you need to use if you have chosen to implement option 1. Compare and contrast the techniques used in option 1 and 2 in terms of the response time perceived by the user, the amount of network bandwidth that can be saved, and the amount of computation being carried out by the cache and the origin server.

1. On the server side, option 2 will be slower than option 1, because it requires server to break a file into fragments, save a list of fragments that can be used to re-construct the file, calculate MD5 hash code for each fragment and save this information somehow somewhere, while option 1 only requires that server to receive and write the file. So the response time will be longer in option 2, the amount of network bandwidth will be discuss later, and the amount of computation will be much more in option 2 than in option 1 (mostly relates to the upload file process).
2. Between the server and the cache, and the amount of network bandwidth that can be saved

in option 2 varies, depending on how similar the pictures are. When client asks for a picture that has never been cached, the cache can possibly be able to reuse some fragments it has, but this depends on how similar the picture client asked for is to the picture cache has cached before, so some network bandwidth could be saved from here. The more pictures cache has cached, the more network bandwidth can possibly be saved in this process. When client asks for a picture that has been cached, cache in option 2 will be slight slower in response time and requires a little bit more network bandwidth for retrieving the list of fragments. However, the difference here is very small since the list file is small. But this also enables client to always download the newest content of a file with a particular name, while cache in option 1 cannot guarantee this.

3. On the cache side, the amount of computation will be much more in option 2 than in option 1, because when cache receive the list, it will compare the list with the list of fragments it has stored to find out what fragments are needed but missing, and then send request to server asking for fragments one by one. Cache also needs to read fragments and re-construct the picture. So, the response time in option 2 will also be longer than in option 1.
4. Between the client and the cache, the network bandwidth will be the same.
5. On the client side, the response time will be shorter or longer in option 2 than in option 1. If the cache can efficiently use fragments it has to construct the picture client asks for before downloading anything from server, the response time can be even much shorter. But it will not be much longer because, again, the list file is a small file anyway. The amount of computation is equal in option 2 and option 1, because it only relates to receive an image file and save to local.