

Homework 5

Due at the beginning of class, November 11, 2004
No late papers accepted

Server Implementation

A new RFC (RFCv3) is posted. It contains an (overly) simple amendment that enables file transfers to be accomplished. Modify your server code with the amendment and test your server by sending an e-Parcel from a sender client to a receiving client.

Synchronization and E-Tags

The RFC states that “Once the e-Stub has been successfully deleted, the e-tag MAY be reused.”

1. Give an example in which the provision leads to incorrect results, i.e., present a sequence of requests from (possibly different) clients that lead to an unexpected outcome. Your answer should highlight the fundamental problem that e-Stubs accesses are concurrent, not synchronized, and therefore inconsistencies can arise.

For the rest of the question, you should view an e-Stub as a shared object on which processes can read (e.g., GET) or write (e.g., ACCEPT).

2. Propose an algorithmic solution to e-Stub consistency. Your solution should allow E-Tags to be re-used. Your solution cannot force a client to run out of e-Stub space on the server and it should be scalable at the server side. Your solution should be completely asynchronous. Your solution should be an algorithm and it is not a protocol.
3. State which consistency models are satisfied by your algorithm, which ones are not, and give a concise explanation for each answer. Additionally, you should explain whether each consistency protocol is important for e-Stubs and why. Include both data-centric and client-centric models.