

196:001 Homework 1 Solution Keys

1. $\text{Prob}(\text{success}) = 1 - (1 - p)^{kT}$.
Substituting $\text{Prob}(\text{success}) = 0.90$, $p = 0.1$, $k = 3$ we get $T = 76$ (approx).
The expected overhead = $k \cdot ((1 - (1 - p)^{T-1})/p + T \cdot (1 - p)^{T-1}) = 52$ (approx)
2. Tapestry has a provision for creating redundant roots of objects. One approach is to add “salt” to the file name (thus to host a file xyz, several roots corresponding to xyz1, xyz2, xyz3 etc will be created, each pointing to the server containing the nearest copy of xyz. It is quite likely that a client will find one such root in its own partition pointing to the nearest replica in the same partition. Also, if A and B are two partitions, and a client in partition A wants to upload a file into the system, then Tapestry has a way of designating a root for that file in the same partition (unlike Plaxton routing that has a rigid method of designating the root -- it may even belong to a different partition B)
3. Unidirectional routing will require 6 steps:
0110 1001 1100 1010 → 1110 1001 1100 1010 → 0010 1001 1100 1010 →
0100 1001 1100 1010 → 0101 1001 1100 1010 → 0101 1001 1100 1110 →
0101 1001 1100 1111

Bidirectional routing will require 3 steps:

0110 1001 1100 1010 → 0101 1001 1100 1010 → 0101 1001 1100 1110 →
0101 1001 1100 1111

Notice that the first step is traveling backwards by subtracting 2^{12} .

4. The motivation behind such a hybrid network should be clear. Unstructured Gnutella-like networks are easy to implement, can detect exact match as well as perform keyword-based search. But it is slow, consumes too much system bandwidth. Structured networks are good at quickly locating the desired object, but it is good at finding exact match only. This if the file is stored under “Hawaii weather” but you try to look for it under “weather Hawaii”, you will not find it. There were several interesting suggestions about the architecture of a hybrid P2P network. One is to continue with a structured network topology and use the search (and join and leave) algorithms for structured networks in case exact match is desired, but use the same topology to perform Gnutella-like search when it is a keyword-based search. Another includes creating smaller structured sub-networks or clouds (possibly dealing with specialized subtopics) connected by an unstructured topology. If you search the web, you can find a couple of other architectures that combines the best feature of both, like

[1] Aameek Singh, Ling Liu: A Hybrid Topology Architecture for P2P Systems. ICCCN 2004: 475-480