CSCI 4770 HWP 2: Distributed Search

Assigned: Tuesday, Feb 5
Due: Tuesday, Feb 26, 11:00AM
(LATE SUBMISSIONS WILL NOT BE ACCEPTED)

In the last home work project, we used peer-to-peer technologies to locate other beacons that are currently online and accessible. We also implemented a simple beacon service that manages *key:value* tuple pair. The next step is to be able to access keys that are available in beacons that are not directly known to the current beacon.

For this project, the beacons from HWP1 will be restricted to manage location information about **two** other beacons. You are free to use any policy to choose a specific set of peers. We extend the services provided by Home Work 1 as follows:

- **searchget(token, searchKey, hopCount)** If the requested key *searchKey* was available in the beacon (because of an earlier SET operation), the corresponding *value* is sent back. If the key *searchKey* was not available, a recursive **searchget** is invoked by this beacon (on behalf of the requestor) on all the beacons that it knows of (restricted to two for this project). Every such forwarding decrements the hopCount. Once the hopCount reaches 0 without successfully finding the file, the system returns an error message.

Note that your implementation might return multiple values for the same key. It might also return an *KeyNotFound* error, even though there is a path available from the source to the destination. As usual, you are free to choose the exact technique to provide the service described above. You could use a traditional RPC style implementation or a multi-way RPC implementation (as discussed in the Active Names paper).

Submission

Please submit your project, along with a succinct report called **REPORT.txt** (plain text is fine) describing your approach, the merits of your approach and compilation instructions. You will turn in your complete project as a single tar file. On gemini, please use `/home/profs/surendar/bin/turnin UBICOMP HWP2 <your tar file>` to submit your assignment. You can submit your assignment multiple times. I will only use the latest submission. To see the files that you had submitted, try `turnin UBICOMP HWP2`. **Remember, I will randomly choose students who will be asked to explain their approach in person**. Issues that you might consider while developing and evaluating your system are:

1. **robustness:** How reliable is your system against failures? Does your beacon recognize forwarding loops? (wherein the requests are forwarded around in a loop without making progress towards the destination)
2. **scalability:** If your beacon suddenly becomes popular because of the files that your provide, how much load can you tolerate before your system crashes? Does your service degrade gracefully? Are you immune to denial-of-service attacks? (wherein, beacons repeatedly open connections to you to prevent you from servicing other, legitimate users)