Efficient searching for information is an important goal in peer-to-peer (P2P) networks. Searching in an unstructured P2P network is particularly interesting due to the random nature of the P2P overlay links. Due to this unstructured nature of the network, providing guaranteed lookups is a hard-to-achieve goal. We propose a probabilistic search mechanism called SPUN, that increases the success rate of queries while keeping the bandwidth consumption considerably low. The core principle of our algorithm is that successful query paths develop during the lifetime of P2P network and these paths converge towards the target objects.

### Types of P2P Networks

- **Unstructured P2P**:
  - Peers are connected to a typically random set of neighbors.
  - Unstructured P2P systems are very efficient for node joins and leaves as they do not need to maintain any particular network architecture.
  - Ex. Gnutella

- **Structured P2P**:
  - Structured P2P networks have well defined neighbor links.
  - Ex: Chord and CAN.
  - CHORD maintains the P2P overlay network as a ring structure CAN maintains it as a d-dimensional toroidal space.
  - These well-defined neighbor links provide guaranteed and bounded-time lookups.

### Related Routing Schemes in P2P

- **Flooding[2]**:
  - This is a breadth-first traversal of the underlying graph where the querying node contacts all the nodes reachable within Time-To-Live (TTL) hops.
  - Simple algorithm
  - Too high traffic overhead

- **Random Walk[2]**:
  - Blind search mechanism where the querying node sends k query messages to k random neighbors.
  - Every node that receives the query forwards it to a randomly selected neighbor.
  - Low success rates

- **APS[1]**:
  - Each peer maintains a local index consisting of one entry for each object it has requested per neighbor.
  - The index value for a given object for given neighbor is the relative probability of that neighbor successfully answering a future query for the given object compared to its neighbors.
  - Searching is initiated by a querying peer by sending k walkers to the subset of its neighbors with highest local index values for the object of interest.
  - APS prevents neighbors marked as bad be queried in future.

### SPUN Data Structures

- **SPUN Index**
  - Each peer to keep a local index of relative probabilities of success per neighbor for each object it requested through that neighbor.
  - Each entry is a vector of relative success rates per neighbor for each object that has been requested through it.
  - The following is an example of a local index maintained by a peer for object 01. (SR is the relative probability of success and SR at each hop distance from peer through given neighbor is maintained)

<table>
<thead>
<tr>
<th>ObjectId</th>
<th>Neighbor</th>
<th>SR(hops=1)</th>
<th>SR(hops=2)</th>
<th>SR(hops=3)</th>
<th>SR(hops=4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>A</td>
<td>30</td>
<td>40</td>
<td>40</td>
<td>50</td>
</tr>
<tr>
<td>01</td>
<td>B</td>
<td>30</td>
<td>20</td>
<td>20</td>
<td>30</td>
</tr>
</tbody>
</table>

- **Peer Profile**
  - Each peer keeps its own profile. Each profile consists of:
    - the object ids of latest n queries that it generated or passed through a hit
    - the object ids of latest n queries that it generated or passed through a miss
  - Each object-id of queries for which a hit/misss was passed through, is associated with a relative probability of success.

### Searching Overview

- Searching is based on simultaneous deployment of k walkers by a querying peer.
- k walkers are selected probabilistically based on the querying peer’s local index.
- Each walker travels along its own path in the network and returns a hit or a miss along the query path back to originator.
- A peer that receive query send a hit back in reverse path if object is found in local storage and terminates query, else query is forwarded to one neighbor.
- If file is not found and TTL has expired, receiver peer returns a miss.
- Each peer sends its profile in any message (query, hit, miss) passed through it to receiver of message.
- Each peer along the query path except one farthest from requestor, also sends their relative probability of success of previous neighbor (i.e. hop=0 value in its index) in query path.
- Each intermediate peer along reverse query path use this information to update its local index.

### References

