Bi-Weekly Update (Feb 21)

Gather information, Download dependencies, Run tests	February 14-21

Overview

During the first two weeks, we focused on gathering information about the different types of algorithms used to search information in unstructured decentralized peer to peer networks, as well as begin to work with NS3 to eventually run our tests.

The sections below detail our work thus far, as well as how we plan to expand in the coming weeks.

Gather Information

We have been gathering information about the 3-4 main search methods we want to study in more detail. We are focusing on the performance of these implementations in a dynamically changing network, specifically their lookup success rate, latency, resilience and efficiency.

Query Flooding:
This method is a simple search algorithm where queries are broadcasted to all nodes in the network. This ensures high recall, but is very inefficient in terms of network bandwidth and time. Due to the exponential growth of the broadcast, scalability and practicality in larger networks [4]. We hope to see how effective the flooding will be when nodes are continuously leaving and joining a network.

Random Walks:
Random walks are essentially a probabilistic search method where queries are passed along a random selection of neighbouring nodes. Due to the nature of this search method, it can reduce network traffic and is more scalable compared to flooding. The drawback to this algorithm is the performance dependency on the topology of the network [5]. We hope to see if the random walk search method will maintain its efficiency with a dynamically changing network.

Hybrid Approach:
Hybrid approaches combine different search strategies, aiming to balance certain strengths of each. This offers a middle ground between exhaustive and probabilistic approaches.

- **Expanding ring search [2]:** Also referred to Iterative deepening, this method the querying node periodically issues a sequence of BFS searching with increasing depth limits. This method is terminated when its maximum depth or the query result hits, making it a good algorithm for instances where the initial number of data items returned is important. Because of this, it can have downfalls in efficiency or success rate based on the network topology.
- **Gossip based search [6]:** This method is a combination of a partial anti-entropy algorithm and rumor based propagation method, and has been shown to improve performance of dynamic

environments. Peers push rumors to randomly selected peers at regular intervals, and also perform pull operations to retrieve missing information. This method is able to maintain consistency in large dynamic networks without excessive bandwidth overhead.

Download dependencies

Under the recommendations of our professor, we have decided to use NS3, or Network Simulator 3. NS3 is a discrete-event network simulator based in C++ which provides the ability to test, analyze, and simulate real networks without having hands on access to real world hardware. For this reason, it is suitable for setting up and creating a small peer-to-peer network.

We have downloaded the NS3 software on our devices and followed the instructions from NS3 documentation [1].

Run Tests

While running our tests we have chosen to track specific metrics that we think will be the most applicable to our proposed research topic.

Metric	Description
Query Success Rate	Measuring if a query has reached its goal as a percentage
Query Failure Rate	Measuring if a query has not reached its goal as a percentage (1 - Query Success Rate)
Total Hop Count	Measuring the total number of Hops performed by a single algorithm.
Average Hop Count	Measuring the mean amount of hops made throughout multiple test runs utilizing the same algorithm. (Number of nodes which the query passes through)
Total Query Latency	Total time taken for a query to be completed
Average Query Latency	The average time for a query to be completed under a standardized set of restrictions.

We would like to point out that these are *proposed metrics*, as we have just begun to understand Network Simulator 3 (NS3) there may be implementation issues and or discrepancies which we missed or not considered at the time of creating this table. Therefore, these metrics are subject to change, if the teaching team and or other students were to advise us to add or remove specific metrics we will follow their recommendations.

Furthermore, as a group we aim to have a modular, reusable, and scalable structure with our designs. Meaning we would like to have a standardized network topology which we would continue to iterate on and further increase complexity. This will likely come with challenges forcing us to start from scratch.

Under these assumptions we have came up with the following [1]

- Predefined Node Amount: to create a consistent environment
- Connection Patterns: utilize different strategies for implementing how nodes will be connected such as Mesh, ring, star, random.
- Uniform Parameters:setting standardized parameters such as bandwidth and latency (provided by ns3)

Also subject to change via recommendations from the teaching team

Moving Forward

At this point in our research and development we have only dipped our toes in NS3, allowing for us to install, experiment with, and test its features. Moving forward from the research phase we will propose and implement a proper network topology and structure along with functional code with supported metrics from our findings thus far.

References

[1]nsnam, “ns-3,” *ns-3*, 2019. https://www.nsnam.org/

[2]J. Wu and X. Li, “Searching Techniques in Peer-to-Peer Networks,” in *Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks*, 2005. doi: https://doi.org/10.1201/9780203323687.ch37.

[3]IBM, “Network topology,” *Ibm.com*, Oct. 24, 2024. https://www.ibm.com/think/topics/network-topology

[4]J. Bo, “Flooding-Based Resource Locating in Peer-to-Peer Networks,” *Lecture Notes in Electrical Engineering*, pp. 671–678, 2011, doi: https://doi.org/10.1007/978-3-642-21697-8_85.

[5]C. Gkantsidis, M. Mihail, and A. Saberi, “Random walks in peer-to-peer networks: Algorithms and evaluation,” *Performance Evaluation*, vol. 63, no. 3, pp. 241–263, Mar. 2006, doi: https://doi.org/10.1016/j.peva.2005.01.002

[6]F. M. Cuenca-Acuna, C. Peery, R. P. Martin and T. D. Nguyen, "PlanetP: using gossiping to build content addressable peer-to-peer information sharing communities," High Performance Distributed Computing, 2003. Proceedings. 12th IEEE International Symposium on, Seattle, WA, USA, 2003, pp. 236-246, doi: 10.1109/HPDC.2003.1210033. keywords: {Peer to peer computing;Data structures;IEEE Membership Directory;Internet;Environmental management;Distributed computing;Computer science;Computational modeling;Virtual prototyping;Collaboration},

.