Understanding Node-Capacitated Networks with Improved Simulations

By Winfred Afeaneku

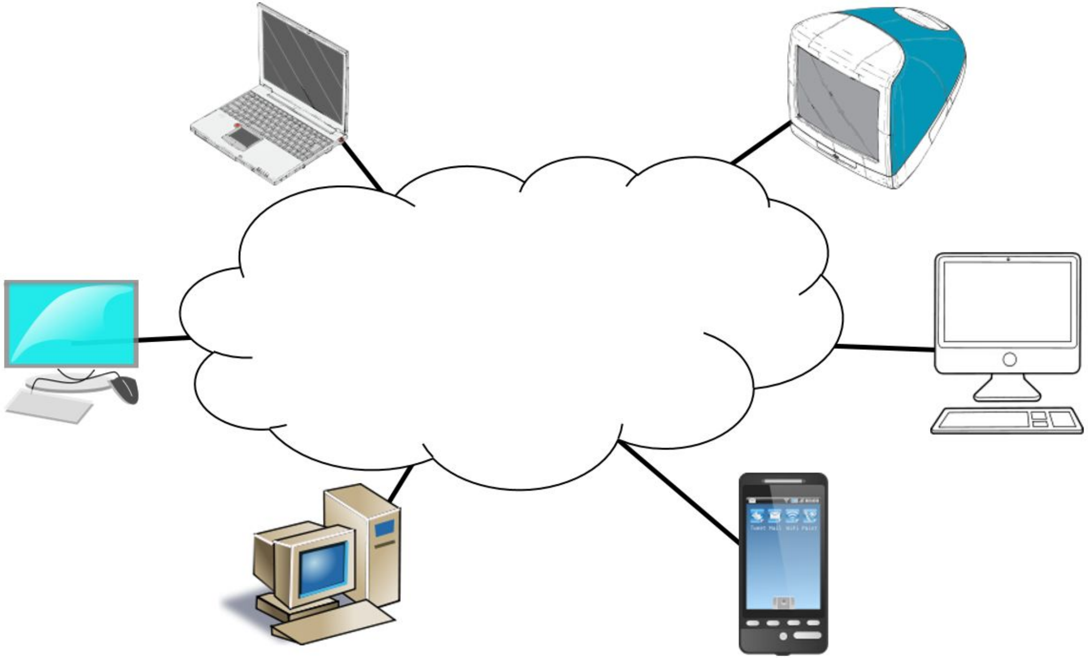
University of Northern Iowa

Outline

- Definition of Overlay Networks: function and importance
- Problem statement
- 2 Simulations
- Result plots
- Future work

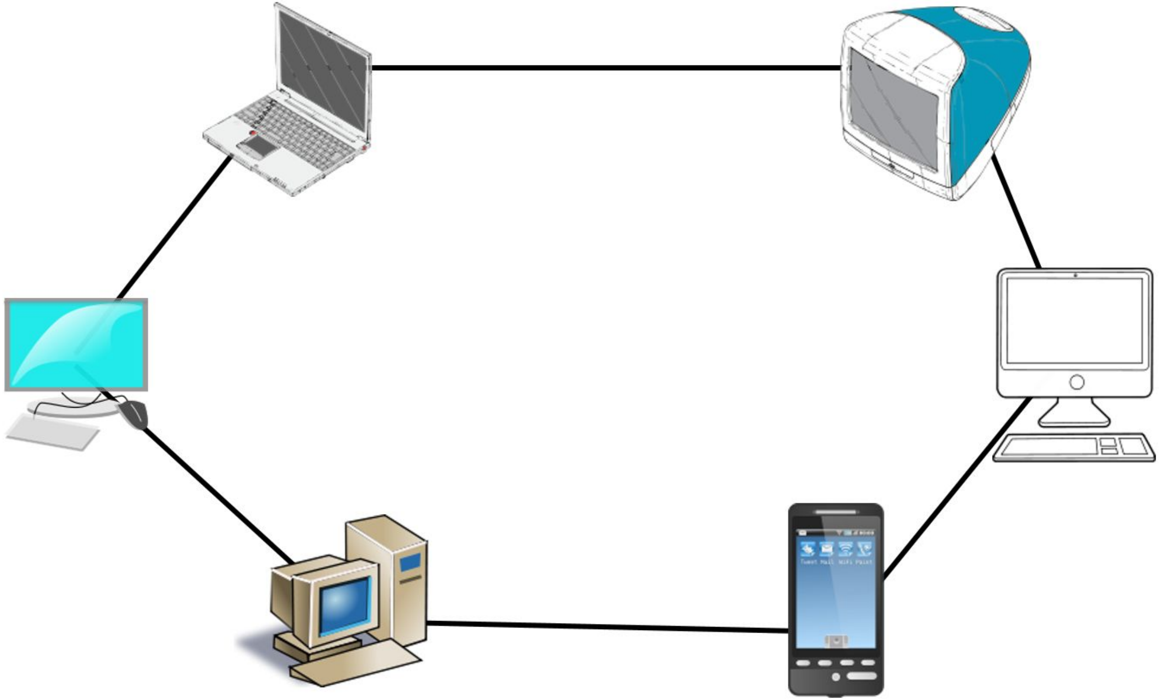
Background

- Overlay Networks



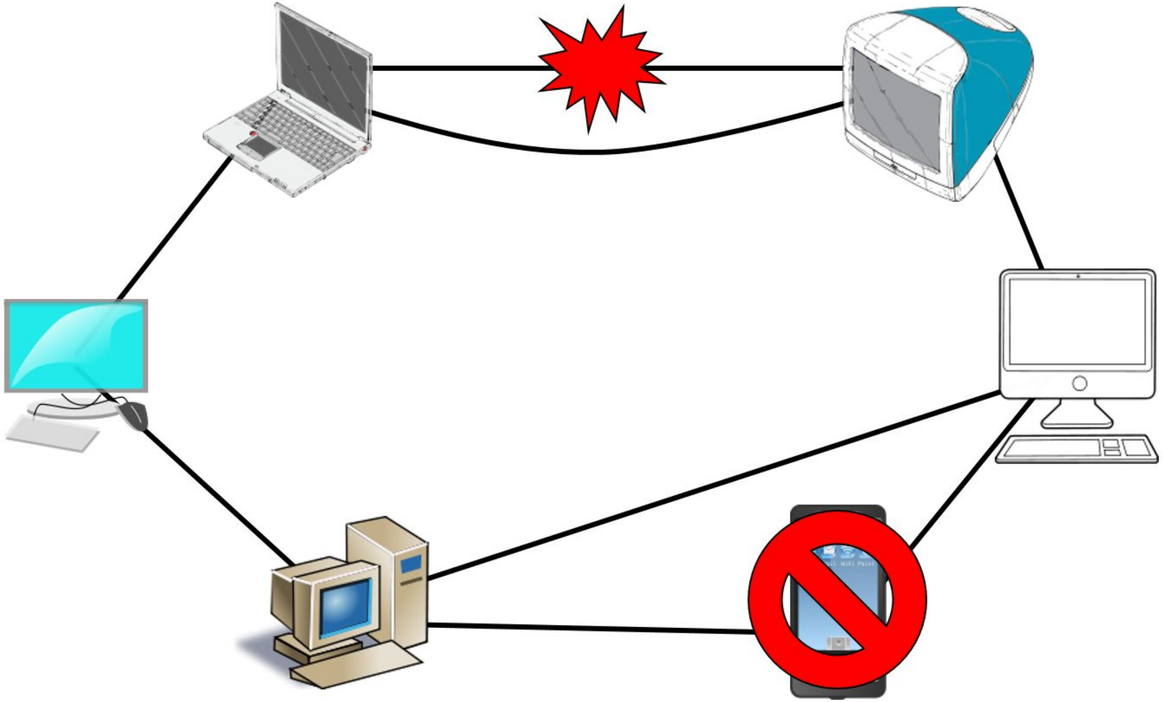
Background

- Overlay Networks



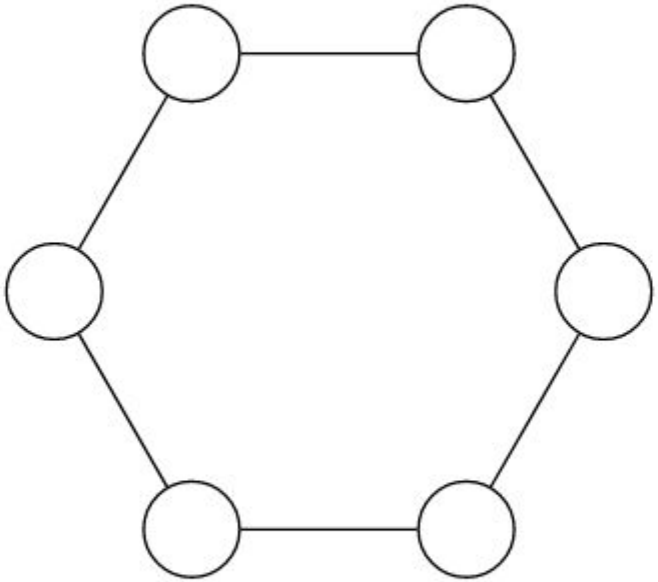
Background

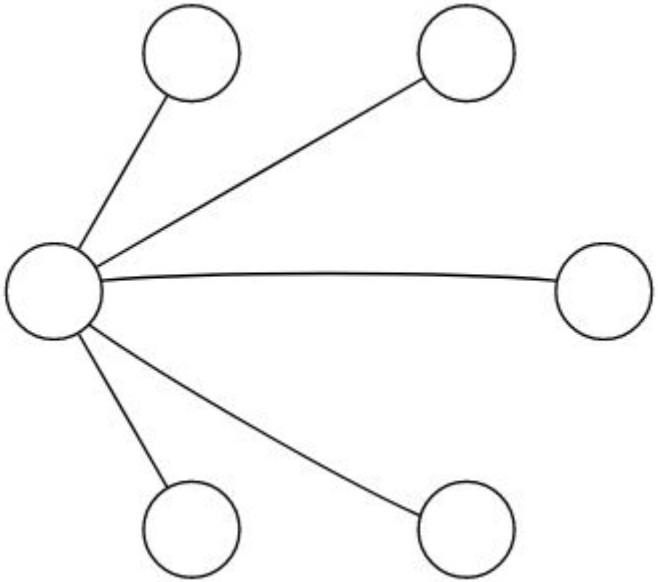
- Self-stabilizing



Background

- Bandwidth limits: logical or physical?





Problem Statement

- *Self-Stabilization in Node-Capacitated Networks:* Can we design efficient self-stabilizing overlay networks when the communication capacities are limited for each node?

- Overlay networks are critical parts of many communication systems, making fault tolerance very important

Python Simulation Parameters

- Network size (*n*): number of nodes in the network

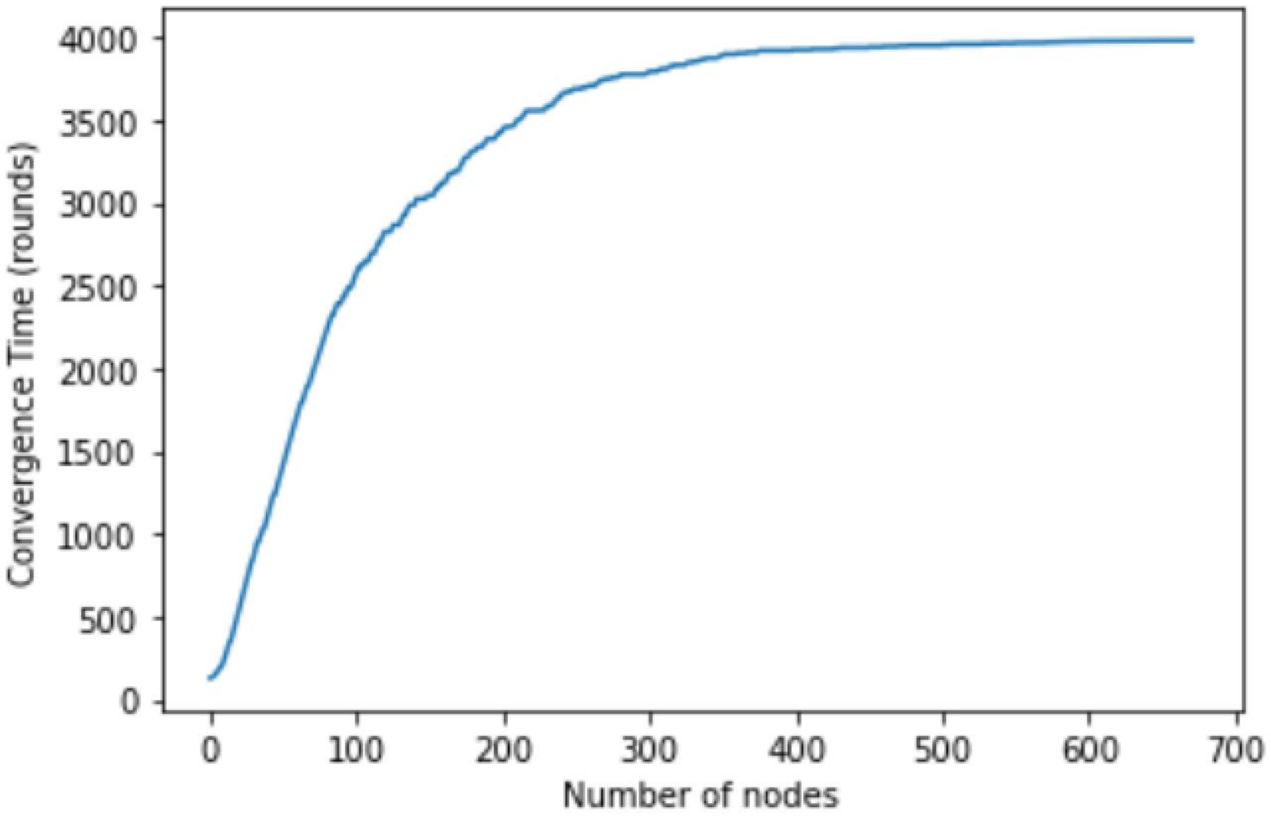
- Goal Topology (clique): every node in the network is connected to the other

- Python Numpy library: Adjacency and Regular Matrices

Simulation

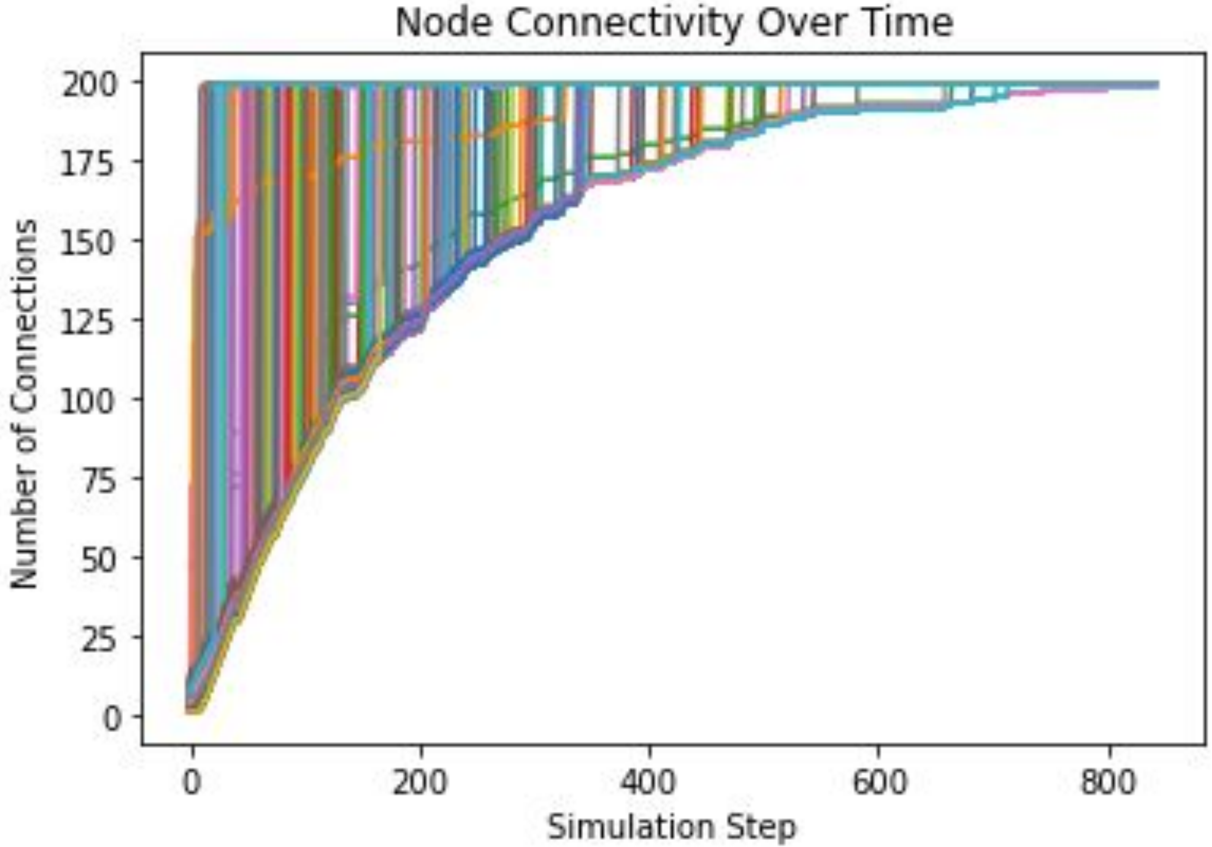
- *Graph creation*:
  - Add edges randomly until we have a connected graph
- *Algorithm Execution* - In each round:
  - A node randomly connects to another another
  - Each message introduces the node to another node, adding an edge (if one does not exist)s
- *Termination*
  - After every simulated round the program checks if a legal configuration (clique) is reached
  - Once the legal configuration is detected, the simulation terminates

Results



Convergence Time (rounds)
4000
3500
3000
2500
2000
1500
1000
500
0
0
100
200
300
400
500
600
700
Number of nodes

Result



Node Connectivity Over Time
Number of Connections
200
175
150
125
100
75
50
25
0
0
200
400
600
800
Simulation Step

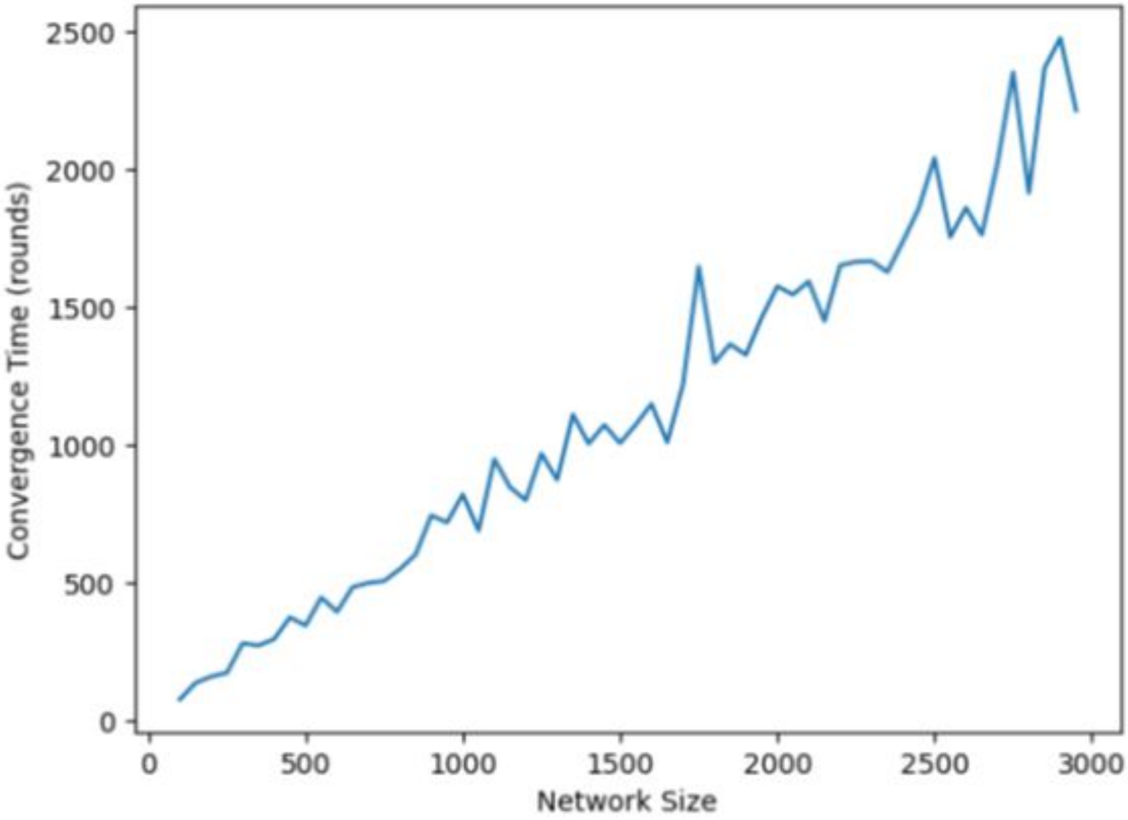
Java Simulation Parameters

- Network size (*n*): number of nodes in the network
- Node capacity (γ): maximum number of messages sent and received
- Message sending limit (σ): maximum number of sent messages

Simulation

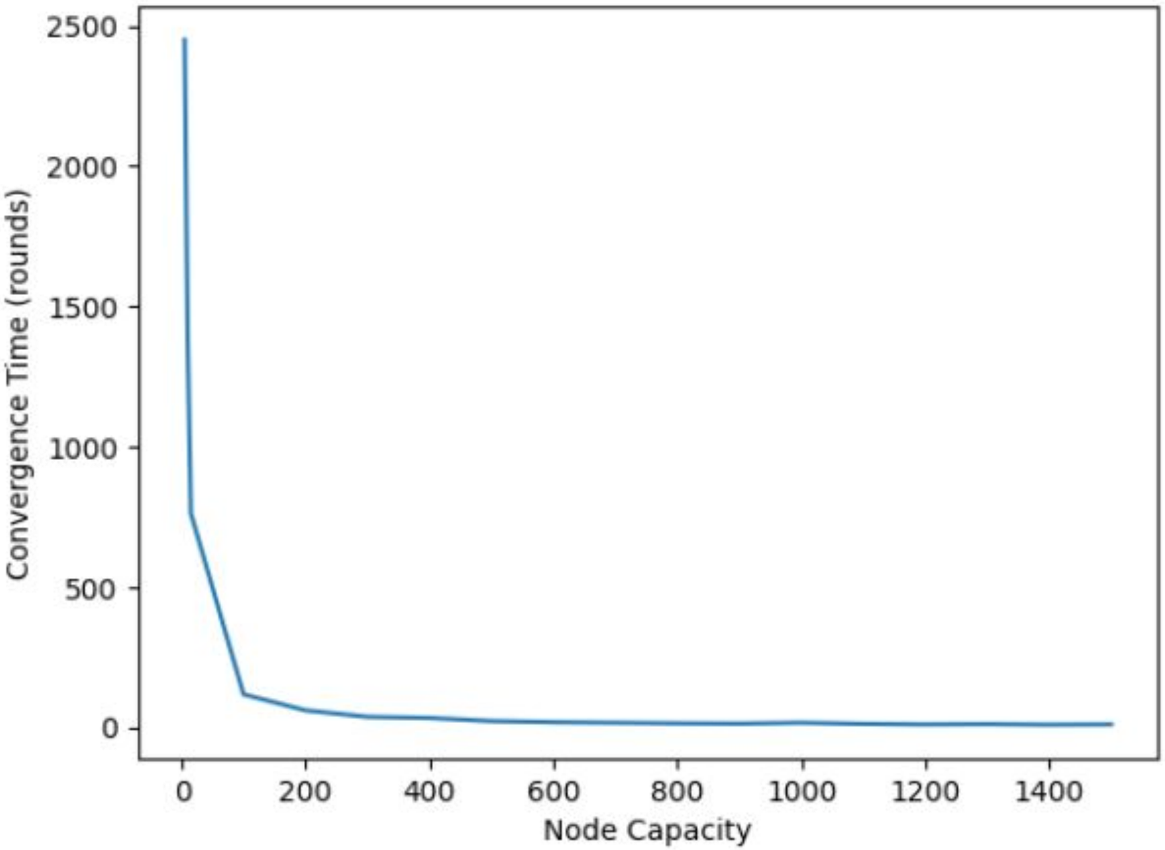
- *Graph creation*:
  - Add edges randomly until we have a connected graph
- *Algorithm Execution* - In each round:
  - A node randomly selects γ messages to deliver
  - Each message introduces the node to another node, adding an edge (if one does not exist)
  - A node creates min(γ, σ) (the minimum of the node capacity and message sending limit) messages and sends them to the appropriate nodes
- *Termination*
  - After every simulated round the program checks if a legal configuration (clique) is reached
  - Once the legal configuration is detected, the simulation terminates

Results



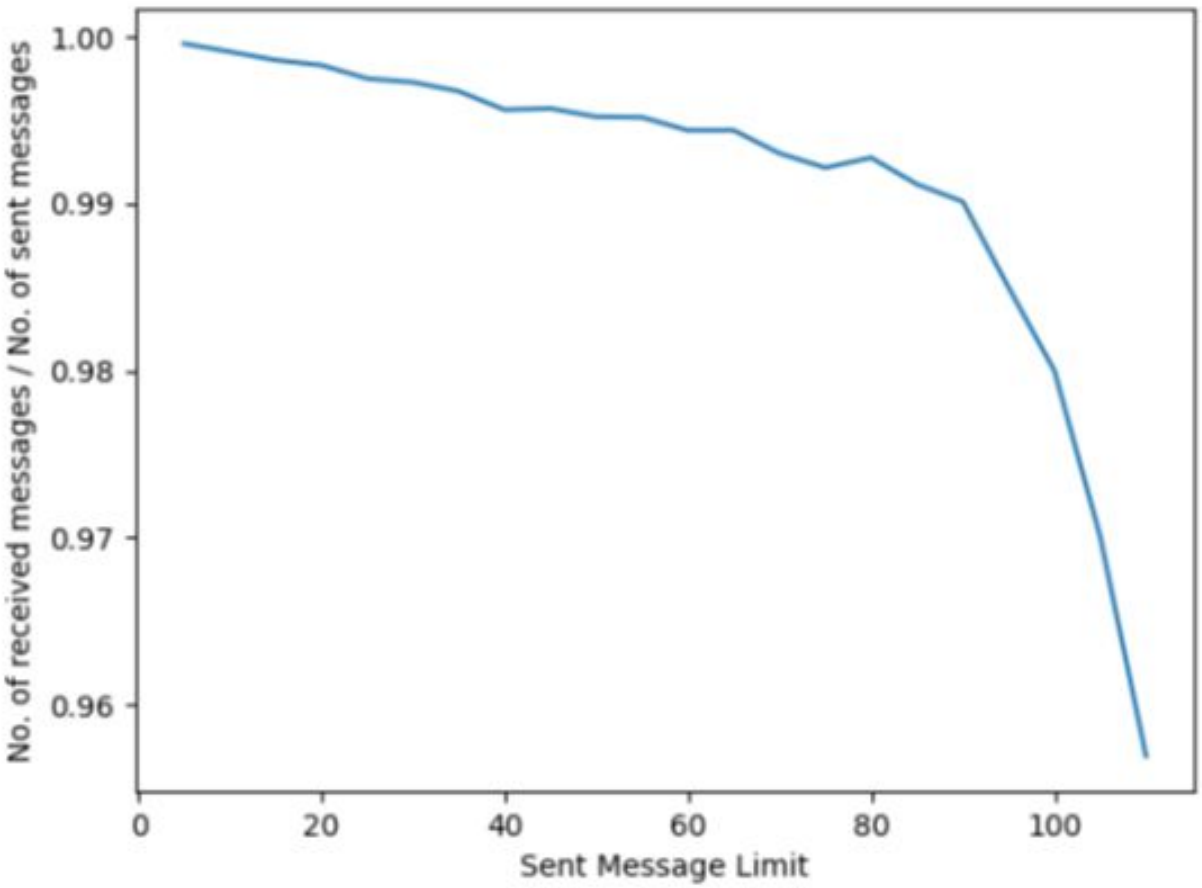
Convergence Time (rounds)
0
500
1000
1500
2000
2500
0
500
1000
1500
2000
2500
3000
Network Size

Results



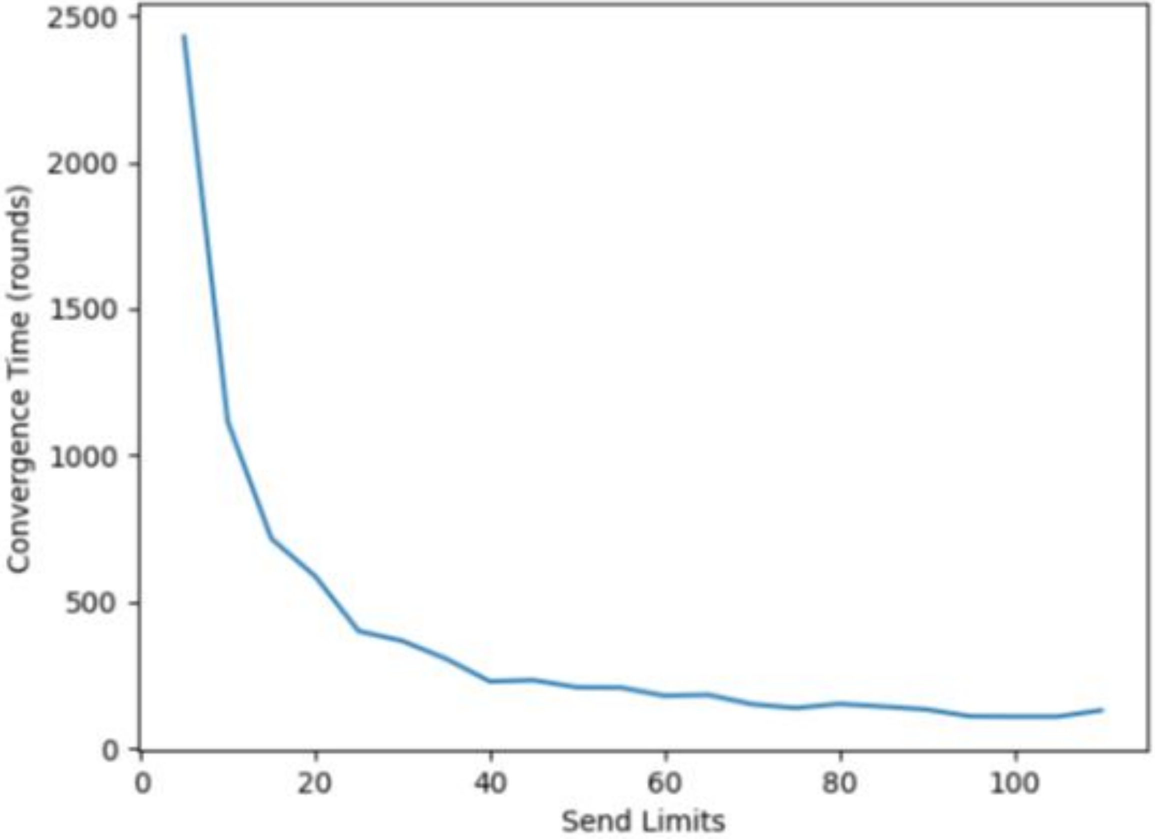
Convergence Time (rounds)
2500
2000
1500
1000
500
0
0
200
400
600
800
1000
1200
1400
Node Capacity

Results



1.00
0.99
0.98
0.97
0.96
No. of received messages / No. of sent messages
0
20
40
60
80
100
Sent Message Limit

Results



2500
2000
1500
1000
500
0
Convergence Time (rounds)
0
20
40
60
80
100
Send Limits

Future Work

- More initial and target topologies
- Simulate millions of nodes
- Create a theoretical framework for node-capacitated overlay networks

The End

Related Work

[2] Aspens, J., Shah, G.: Skip graphs. In: SODA ’03: Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms. pp. 384–393. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2003)
[3] Augustine, J., Ghaffari, M., Gmyr, R., Hinnenthal, K., Scheideler, C., Kuhn, F., Li, J.: Distributed computation in node-capacitated networks. In: The 31st ACM Symposium on Parallelism in Algorithms and Architectures. p. 69–79. SPAA ’19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3323165.3323195, https://doi.org/10.1145/3323165.3323195
[5] Feldotto, M., Scheideler, C., Graffi, K.: Hskip+: A self-stabilizing overlay network for nodes with heterogeneous bandwidths. In: 14-th IEEE International Conference on Peer-to-Peer Computing. pp. 1–10 (2014). https://doi.org/10.1109/P2P.2014.6934300
[6] Jacob, R., Richa, A., Scheideler, C., Schmid, S., T¨aubig, H.: A distributed polylogarithmic time algorithm for self-stabilizing skip graphs. In: PODC ’09: Proceedings of the 28th ACM symposium on Principles of distributed computing. pp. 131–140. ACM, New York, NY, USA (2009). https://doi.org/http://doi.acm.org/10.1145/1582716.1582741
[7] Kniesburges, S., Koutsopoulos, A., Scheideler, C.: Re-chord: a self-stabilizing chord overlay network. In: Proceedings of the 23rd ACM symposium on Parallelism in algorithms and architectures. pp. 235–244. SPAA ’11, ACM, New York, NY, USA (2011). https://doi.org/10.1145/1989493.1989527, http://doi.acm.org/10.1145/1989493.1989527
[8] Onus, M., Richa, A.W., Scheideler, C.: Linearization: Locally self-stabilizing sorting in graphs. In: ALENEX. SIAM (2007)

Background and Motivation

- Pioneering a new and improve way of structuring algorithm that lead to optimal topology
    - Building a more-realistic model for self-stabilizing overlay networks
- Overlay Networks is big contributor to communication in this world

Question they may ask

-