Adapting Peer-to-Peer Topologies to Improve System Performance

Presented by
Chi-Hung Chao

Outline

- Introduction
- Topology Evolution Algorithm
- Balancing Bias Algorithm
- Experiment
- Analysis and Observation
- Conclusion and Discussion

Introduction

- Improving the performance of unstructured Peer-to-Peer file sharing system.
- Controlling the routing protocol and network topology simultaneously.

Introduction

- Flooding based search algorithms are very sensitive to the number of edges in the network graph.

Topology Evolution Algorithm

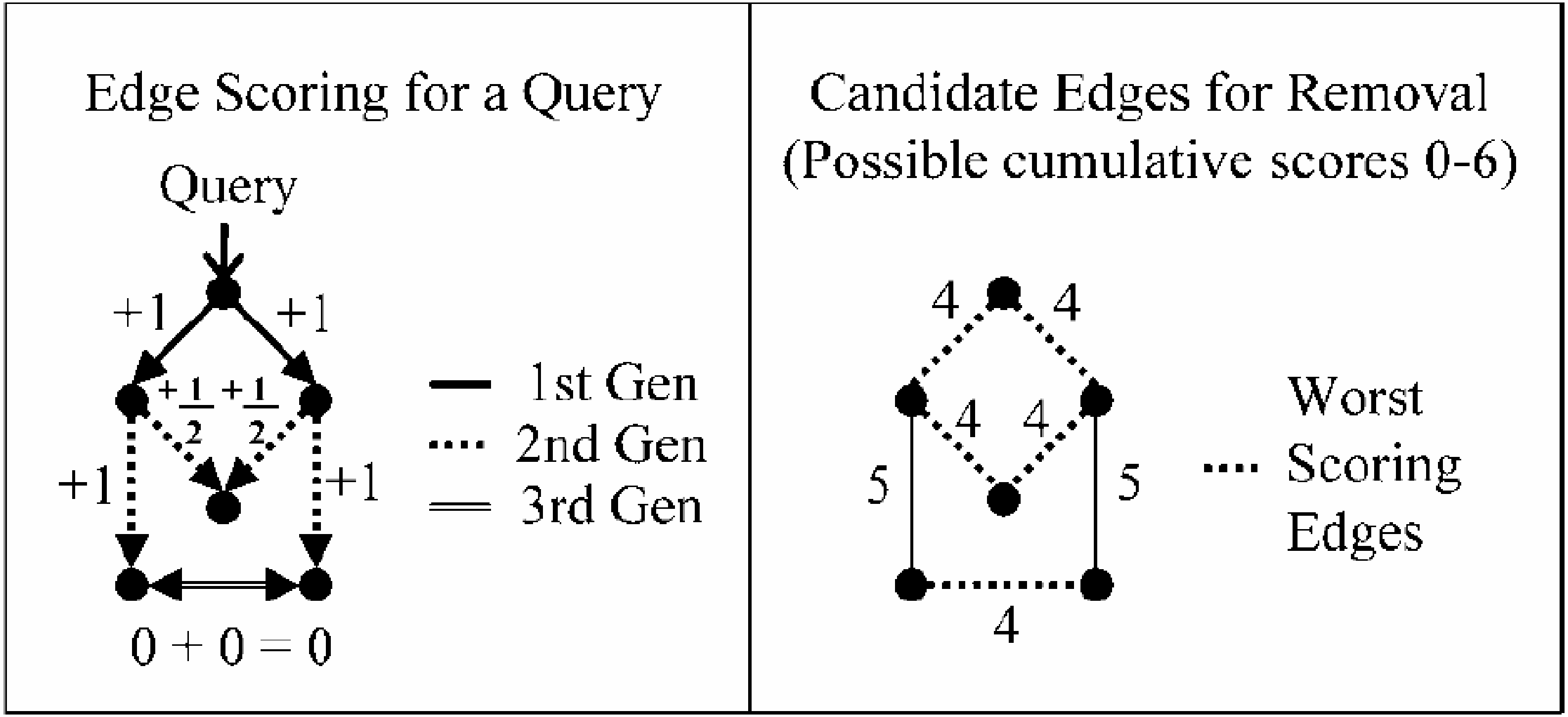
- Edge Thinning
  - Reducing the total number of links
- Diameter Folding
  - Adding some useful links

Topology Evolution Algorithm

- Edge Thinning
  - Reducing search cost by removing the most redundant links without disconnecting the graph.
  - Assigning a “score” to each link within the network to determine how useful it was across all searches.

Topology Evolution Algorithm

- Edge Thinning



Topology Evolution Algorithm

- Edge Thinning
  - Late message
    - Redundant messages received at a later time.
  - Even-length cycles also result in simultaneous redundant message delivery, but no late message are passed.
  - Some bandwidth is used without any benefit to search in these odd-length cycles.

Topology Evolution Algorithm

- Edge Thinning
  - This calculation-removal cycle can be repeated until the graph becomes disconnected, at which point the last edge removed can be re-inserted.

Topology Evolution Algorithm

- Diameter Folding
  - After having created a way to remove the most redundant links, we began considering how and when we might add useful links.

Topology Evolution Algorithm

- Diameter Folding
  - By breadth-first search protocol, the worst-case time for a search to reach all other nodes is the same as the graph diameter.
  - One way to add shortcut links between all node-pairs whose minimum path length is the diameter.

Topology Evolution Algorithm

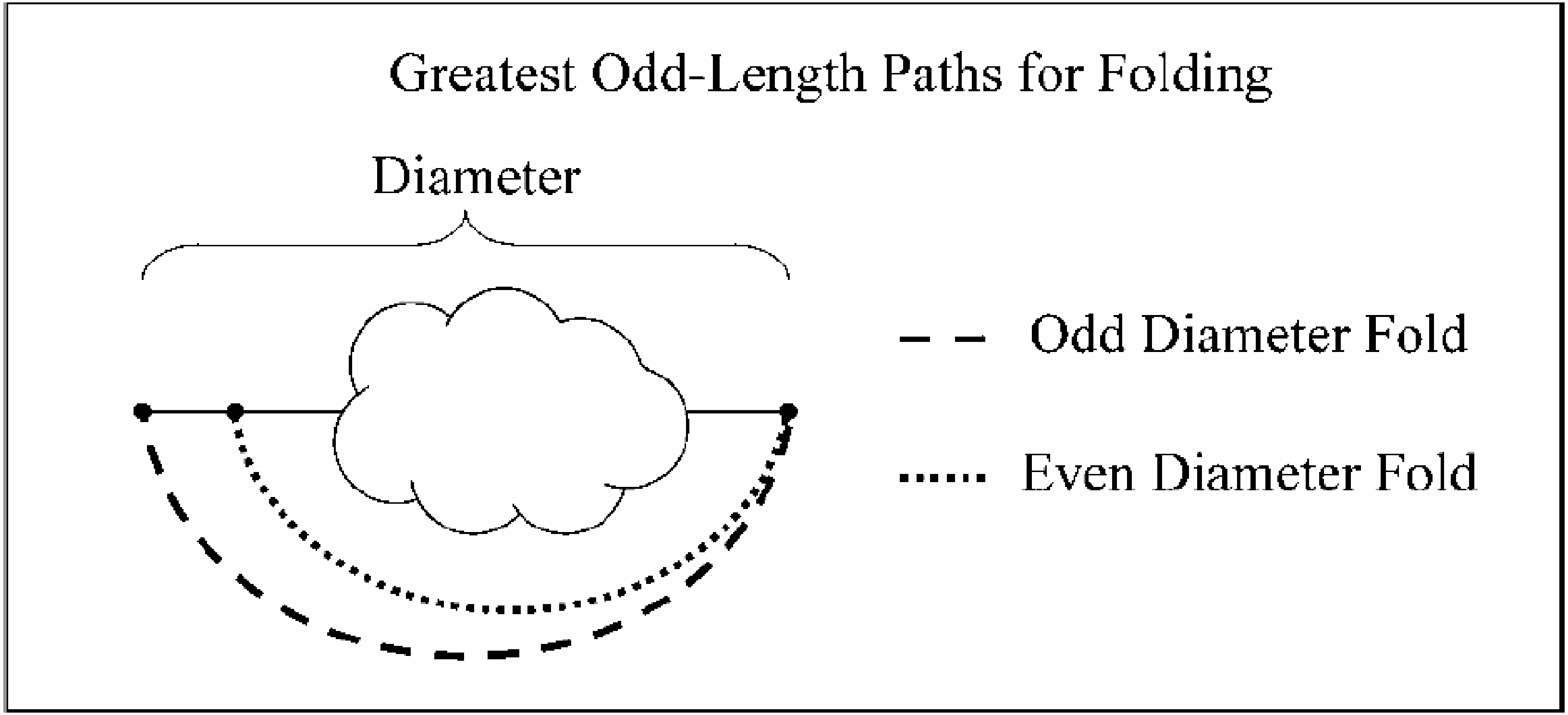
- Diameter Folding
  - So we perform the Diameter Folding algorithm, and then perform thinning from the folded topology.
  - But an edge would often be added that was immediately removed by the next thinning step because the folding step was creating an odd cycle within the graph.

Topology Evolution Algorithm

- Diameter Folding
  - Ensuring that the new cycle that is being created is always of even length
    - Folding operation selects a pair of nodes at the diameter and connects them directly if the diameter is odd.
    - If the diameter is even, connects one of them to a neighbor along the path.

Topology Evolution Algorithm

- Diameter Folding



Balancing Bias Algorithm

- To balance nodes degrees
- In thinning
  - Remove the poor scoring links between nodes with high degrees.
- In Folding
  - Add links between distant nodes with low degree.

Experiments

- Experiment 1
  - Fixed number of links.
- Experiment 2
  - Allowed the number of links to increase or decrease.

Experiment 1

Graph Size		Target Mean Degree	Actual Mean Degree	Initial Degree Statistics			Degree statistics after 5000 steps of alternating thin and fold: Without Balancing Bias			Degree statistics after 5000 steps of alternating thin and fold: With Balancing Bias		
Nodes	Links			Min	Max	Std	Min	Max	Std	Min	Max	Std
32	66	5	4.1	1	8	2.0	2	17	4.4	2	8	1.4
64	178	6	5.6	1	10	2.1	3	33	7.6	2	9	1.7
128	455	7	7.1	2	15	2.8	3	54	9.7	4	15	2.0
256	1020	8	8.0	1	17	2.9	4	59	10.5	4	17	2.7
512	2371	9	9.3	1	20	3.1	5	37	5.6	6	21	2.3

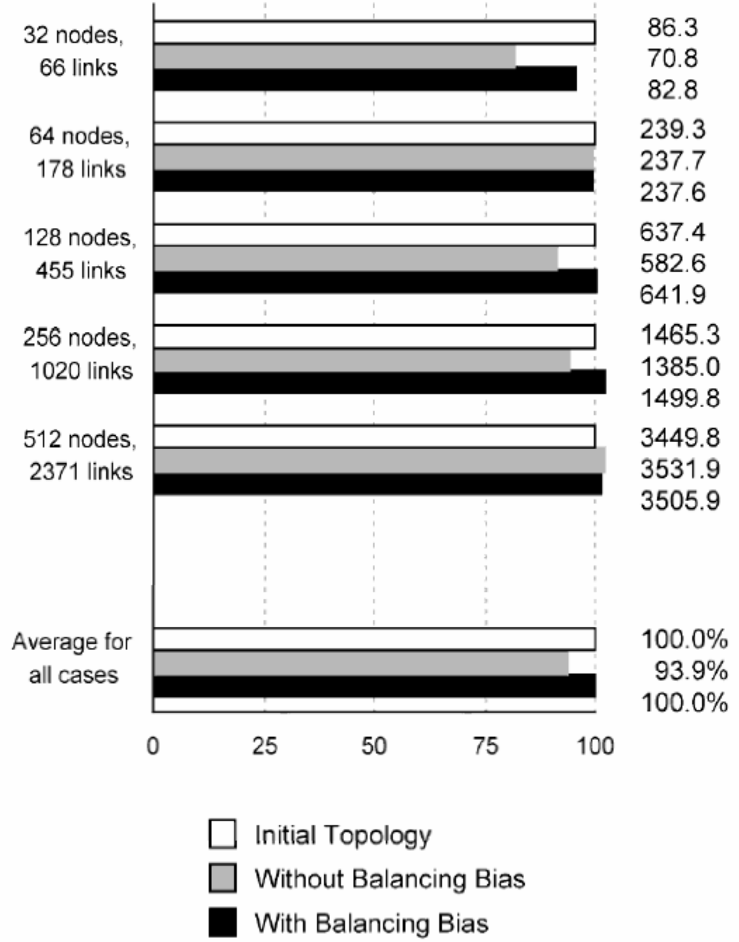
NOTE: Random graphs generated by probabilistically deciding whether to include each possible link, with the probability determined by target mean degree. Here, the target degrees mimic a base two hypercube of the same size.

Experiment 2

Initial Graph Topology	Degree			Path			Search		
	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max
Random Regular: degree 3	3	3.0	3	1	5.1	9	219	226.9	235
	1	3.0	5	1	4.6	7	213	222.6	233
Random Regular: degree 7	7	7.0	7	1	2.7	4	594	634.5	666
	1	3.0	7	1	4.2	7	210	221.9	232
Random Net	1	7.1	13	1	2.7	5	618	645.6	673
	1	3.0	13	1	3.8	7	218	225.9	233

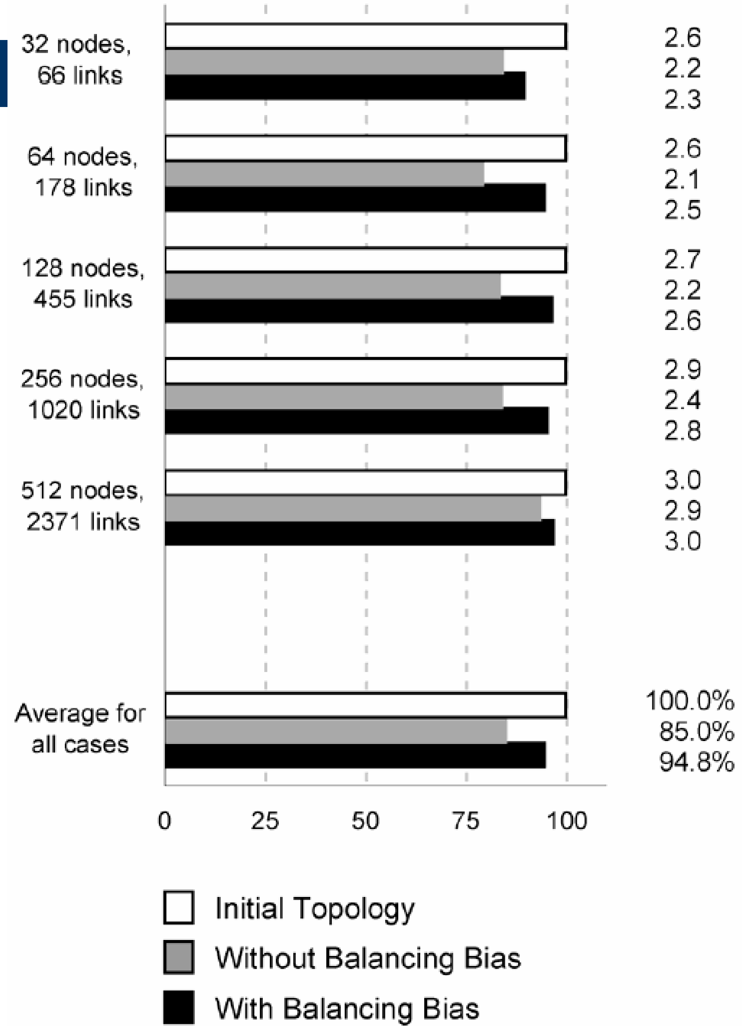
Analysis and Observations

- Bandwidth Usage



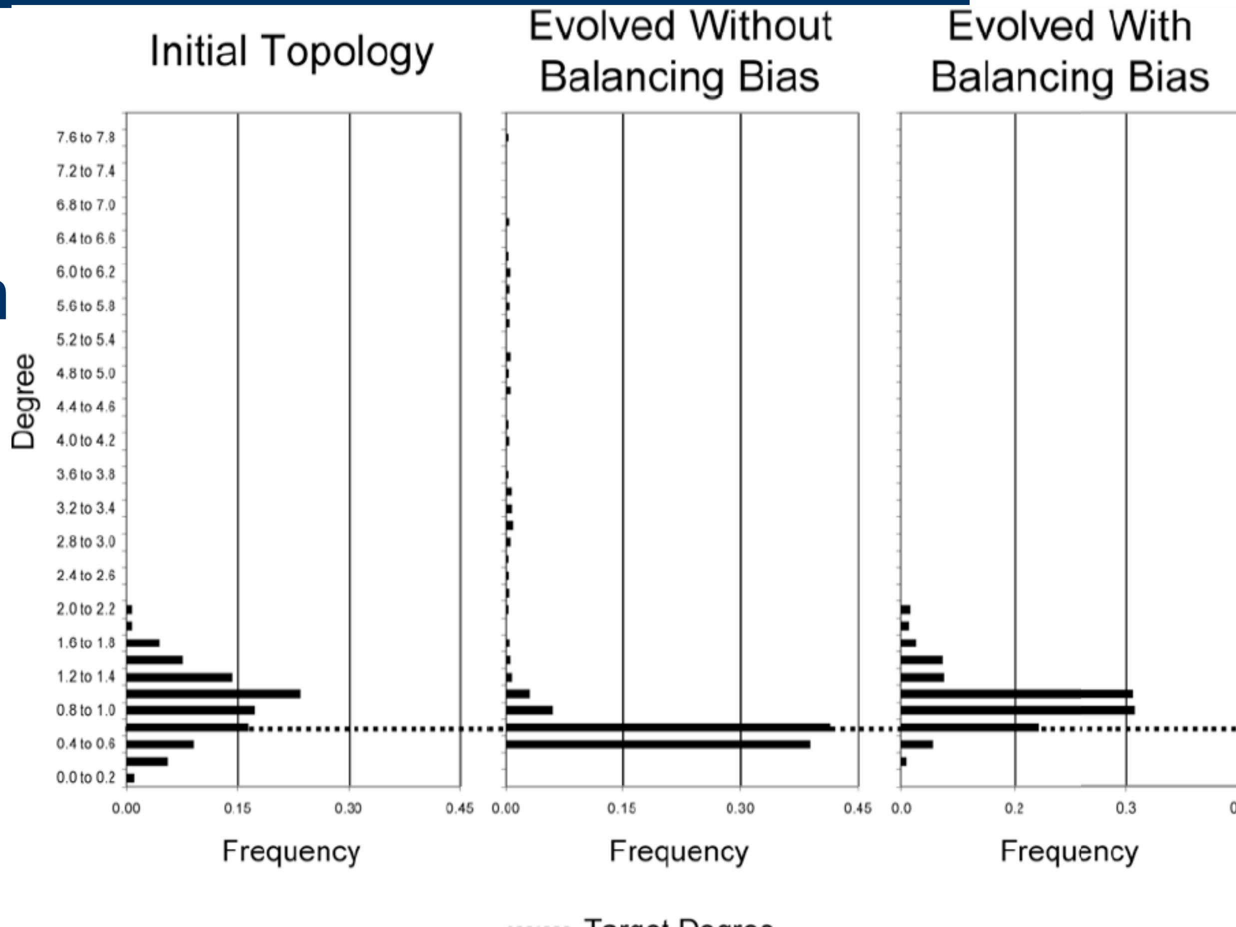
Analysis and Observations

- Search Time



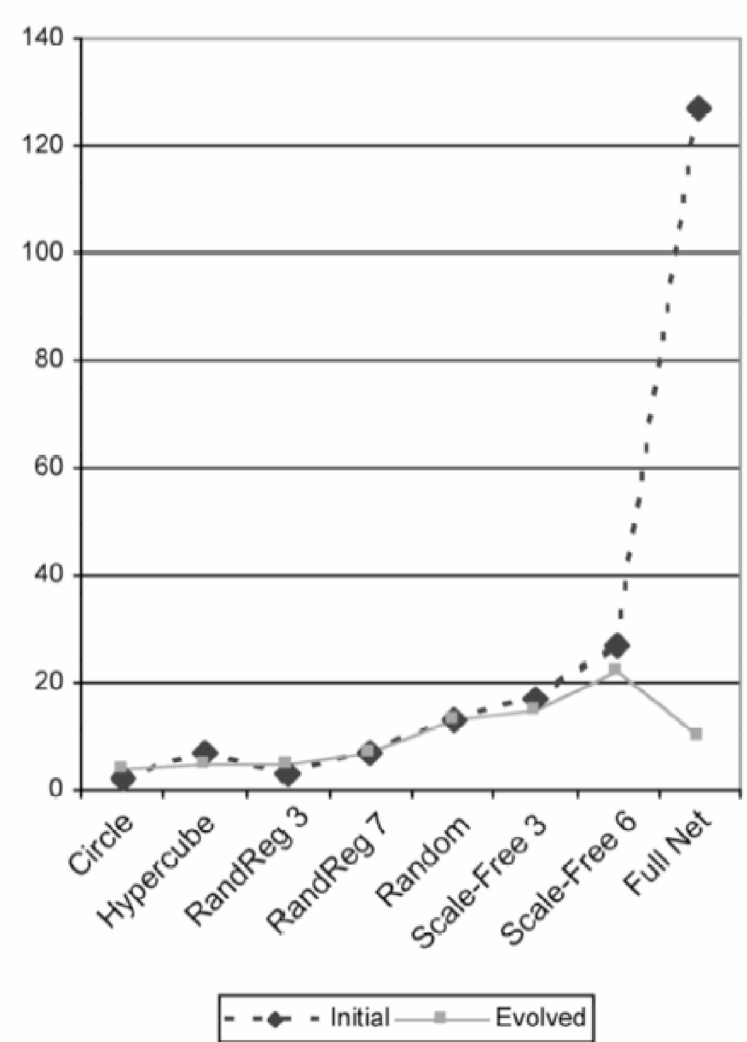
Analysis and Observations

- Average Degree Distribution



Analysis and Observations

- Maximum Degree



Conclusions and Discussions

- These algorithm will lead to network performance improvements.
- But the algorithm need to be performed in a stable environment, and the bandwidth consumption of performing this algorithm maybe large.