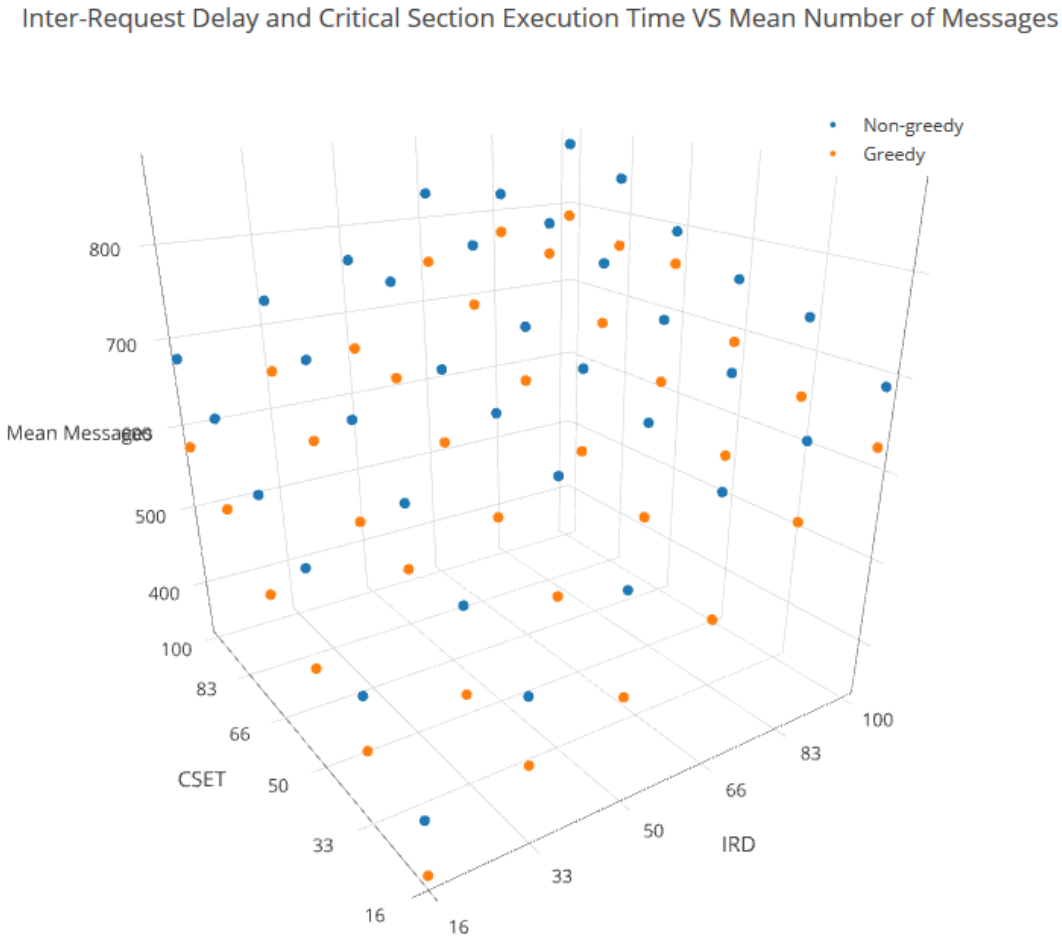
Advanced Operating Systems Final Project
Evaluation

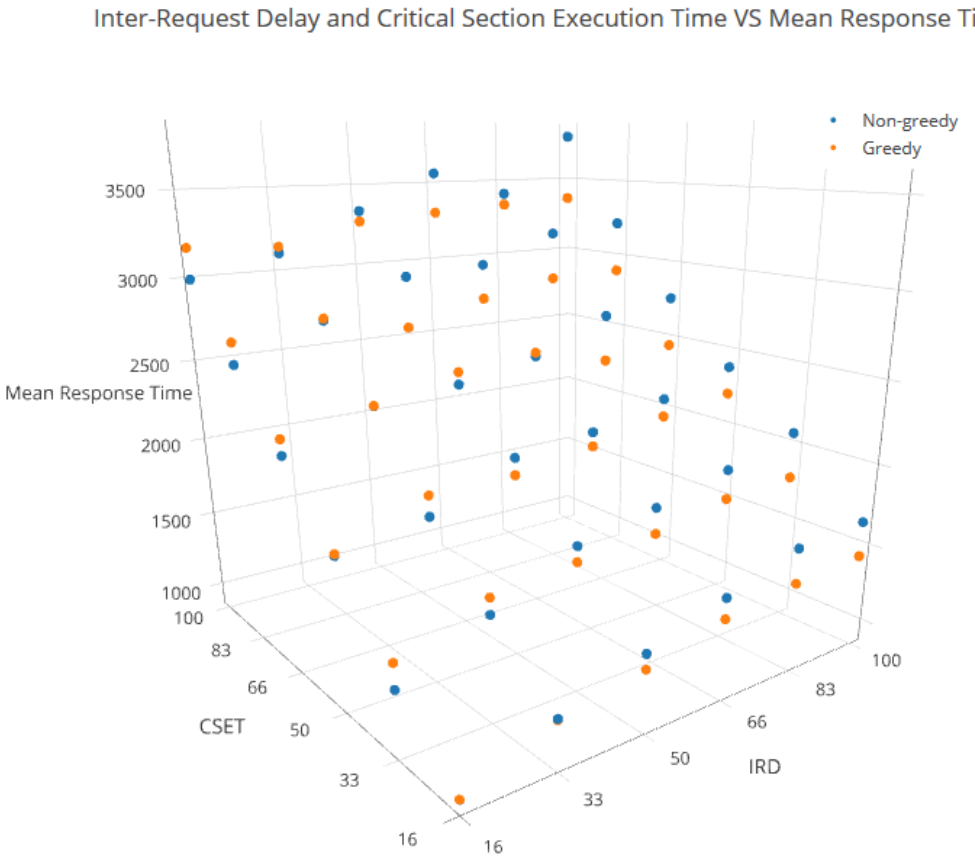
Samuel Chen
sxc165830@utdallas.edu
Tristan Duckworth
txd123130@utdallas.edu
Sabarish Nadarajan
sxn164530@utdallas.edu

August 3, 2017

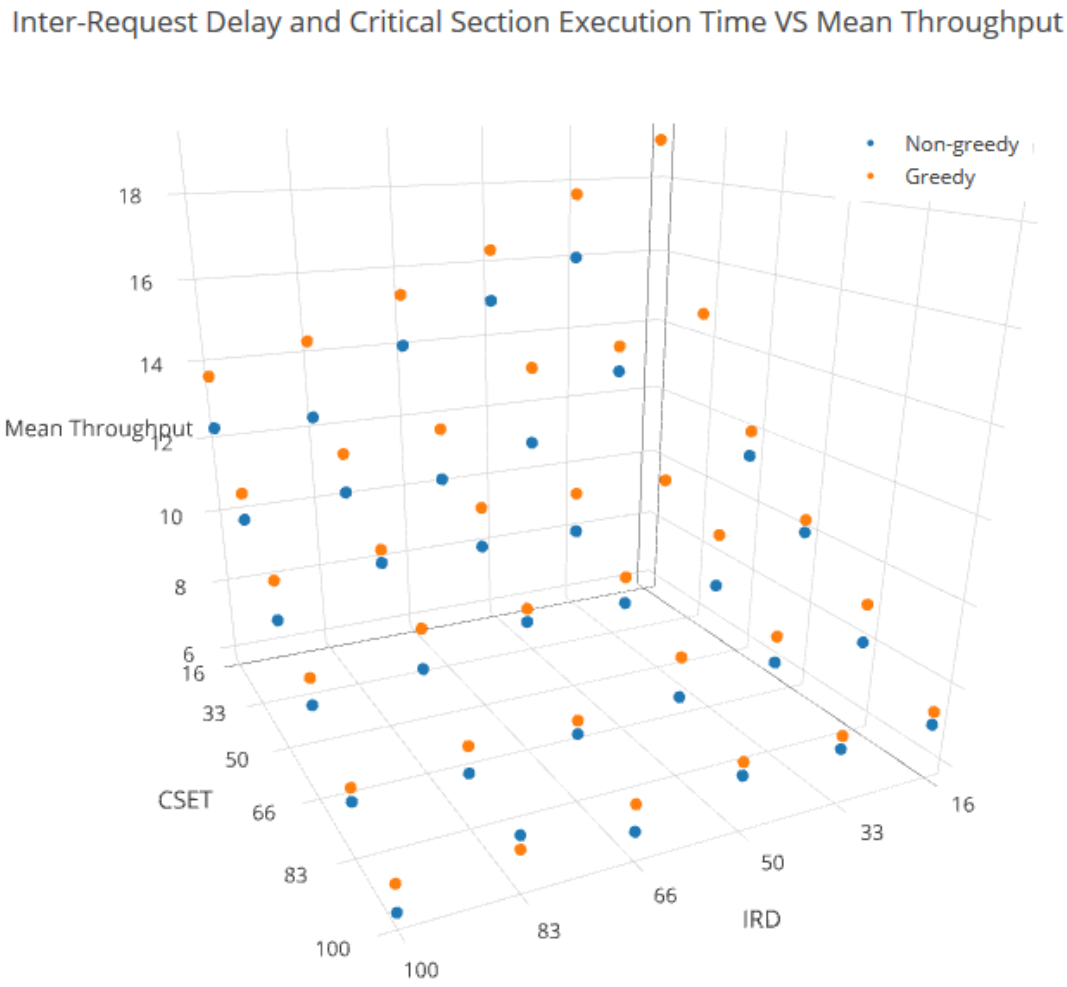
1 Plotting Both IRD and CSET Against Metrics



The above plot shows the relationship between the two input parameters, inter-request delay (IRD, in milliseconds) and critical section execution time (CSET, in milliseconds), and the number of messages sent throughout the network. These tests were run on a perfect 31-node binary tree, with the token starting at the root of the tree. Each data point represents the average of three separate runs with the same configuration. Greedy and non-greedy runs are plotted in orange and blue, respectively. In general, the greedy version outperforms the non-greedy version on this metric, with both seeing a linear increase in the number of messages sent as either IRD or CSET are increased. For an interactive version of this plot, visit this webpage.



The above plot shows the relationship between IRD, CSET, and the response time in milliseconds experienced by applications using the system. These tests were run on a perfect 31-node binary tree, with the token starting at the root of the tree. Each data point represents the average of three separate runs with the same configuration. Greedy and non-greedy runs are plotted in orange and blue, respectively. For IRD values above 50, the greedy version tends to be superior, with the non-greedy version performing better for low IRD values. The CSET value dominates the relationship with the response time in a linear relationship, while increasing IRD values only slightly impact this metric. For an interactive version of this plot, visit this webpage.



The above plot shows the relationship between IRD, CSET, and the network throughput in requests satisfied per second. These tests were run on a perfect 31-node binary tree, with the token starting at the root of the tree. Each data point represents the average of three separate runs with the same configuration. The greedy version dominates the non-greedy version on this performance measure. Note that the axes, from this perspective, decrease as they move away from us. As CSET decreases, network throughput increases linearly. Decreasing the IRD has a similar effect. For an interactive version of this plot, visit this webpage.

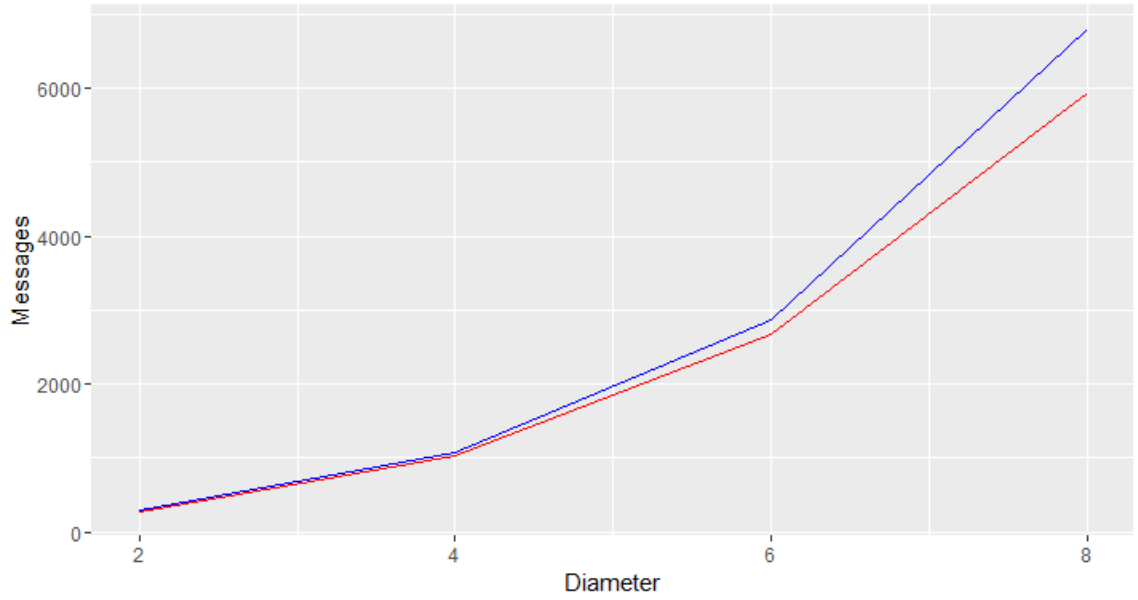
2 Plotting Diameter/Number of Nodes Against Metrics

All graphs are plotted with the following fixed parameters:

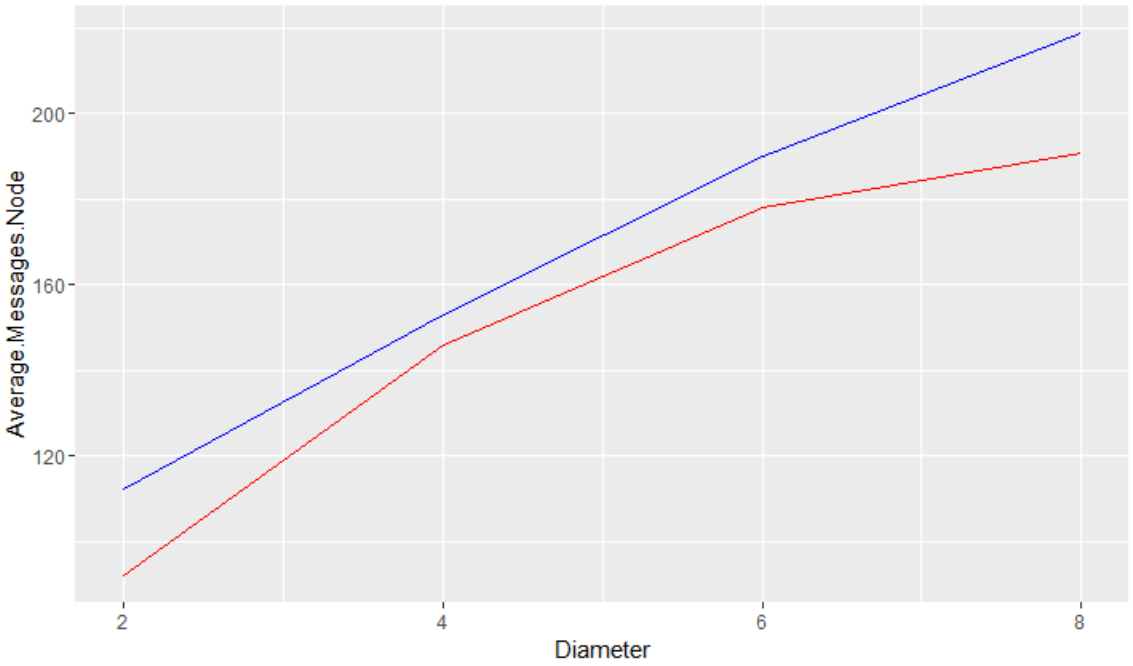
- Inter Request Delay: 20ms
- CS Execution time: 5ms

- The topology follows a set of perfect binary trees with increasing number of nodes (config files attached: config_binary3, config_binary7, config_binary15, config_binary31)
- Each point in the graph represents an average of 5 runs for each greedy and non-greedy algorithm.
- The Red line indicates greedy algorithm execution.
- The Blue line indicates Non-Greedy algorithm execution.
- (Other results attached with CSV files: Results for Graph.csv, Results of execution.csv)

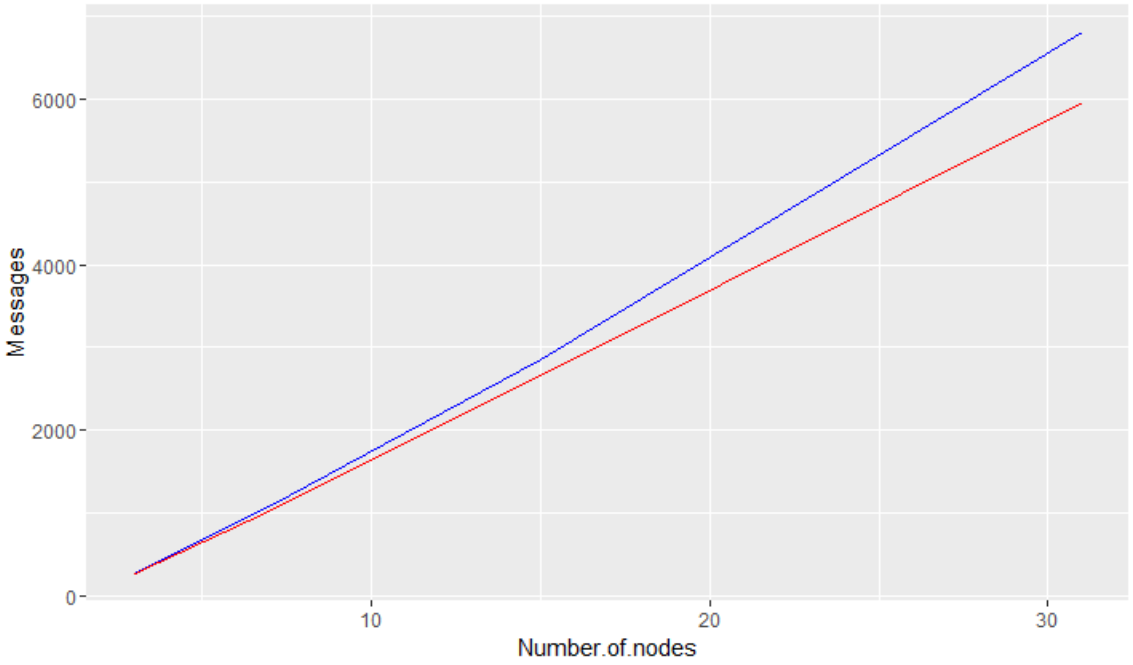
Plot 1 represents a comparison of greedy and non-greedy algorithms for the two parameters: Diameter and the Total number of messages.



Plot 2 represents a comparison of greedy and non-greedy algorithms for the two parameters: Diameter and the Average number of messages per node.

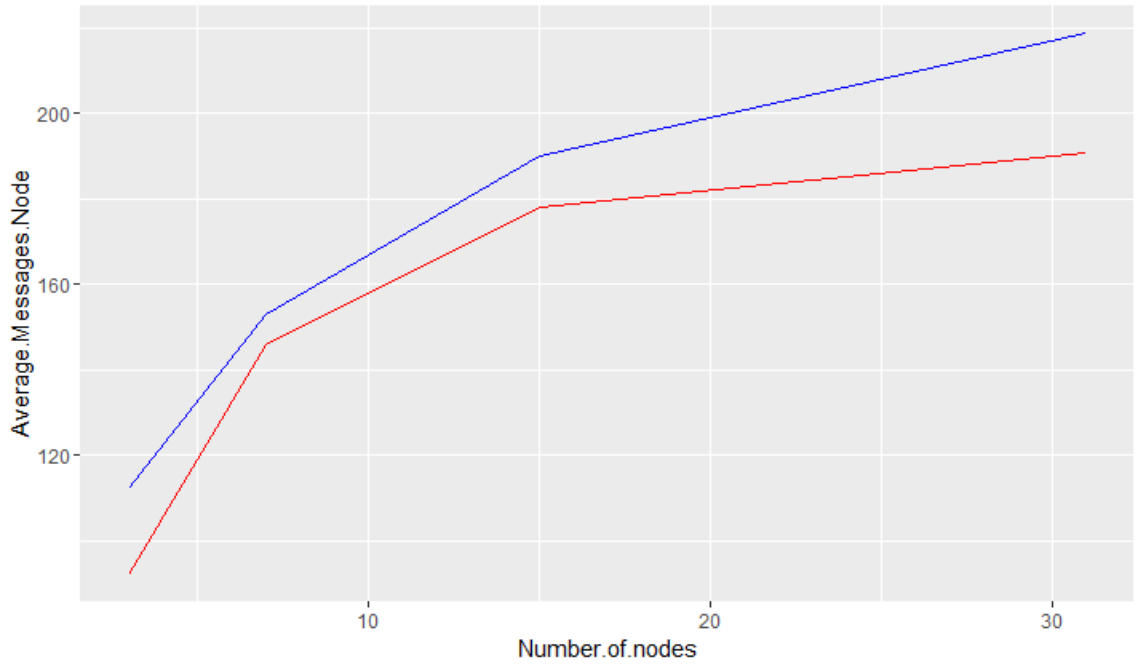


Plot 3 represents a comparison of greedy and non-greedy algorithms for the two parameters: Total Number of nodes and the Total number of messages.

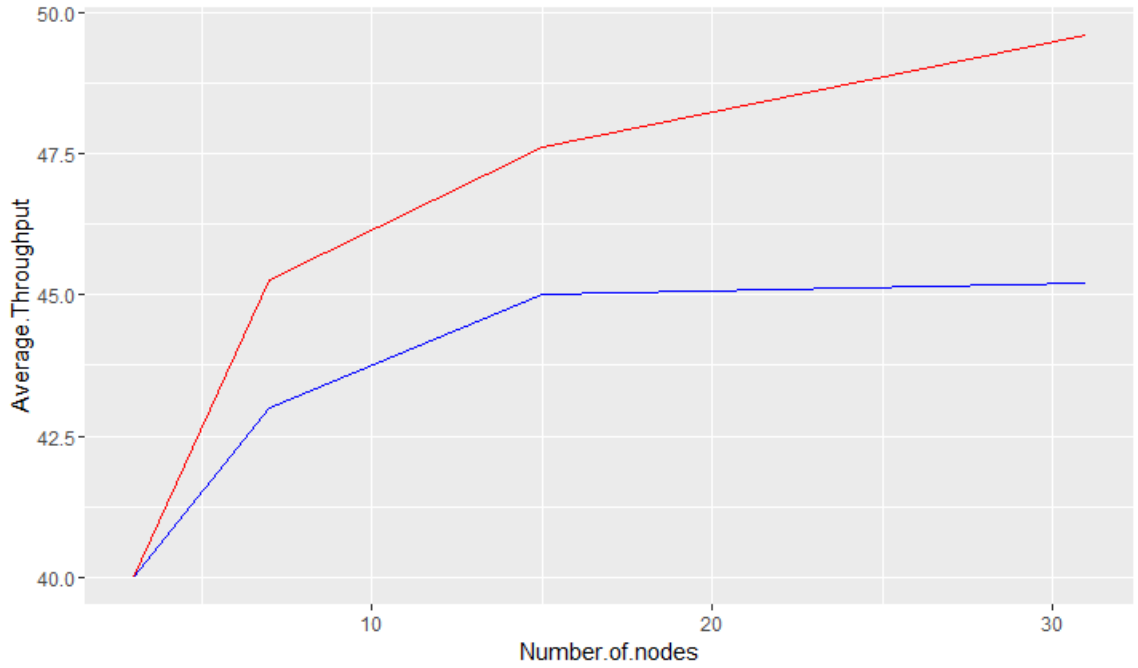


Plot 4 represents a comparison of greedy and non-greedy algorithms for the two parameters:

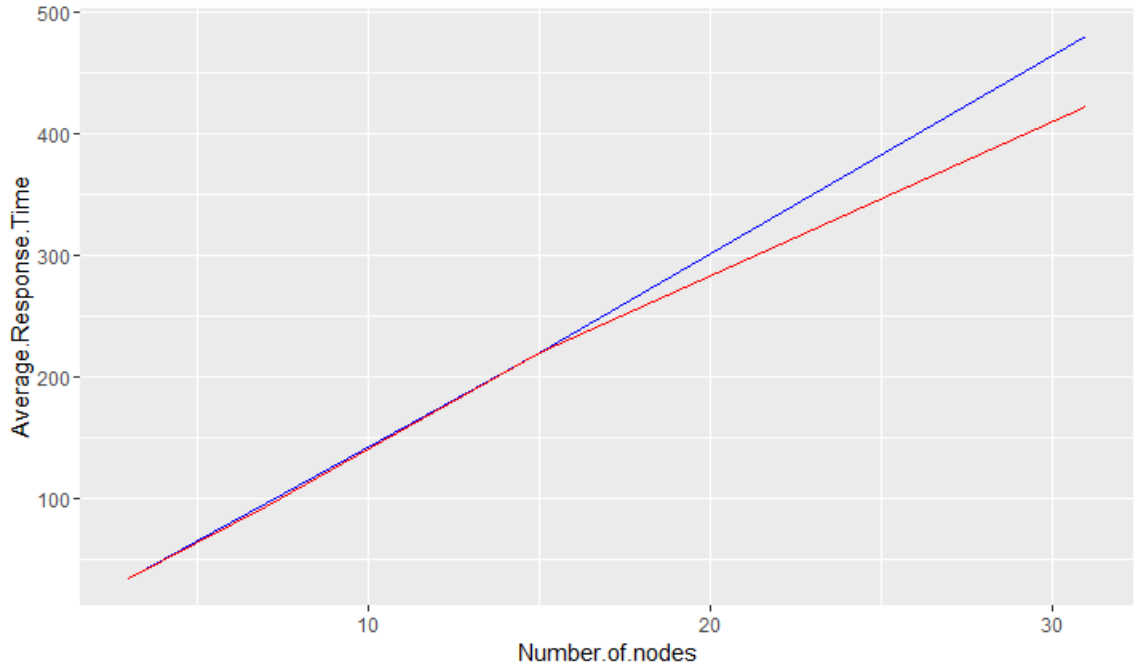
Total number of nodes and the Average number of messages per node.



Plot 5 represents a comparison of greedy and non-greedy algorithms for the two parameters: Total number of nodes and the Average throughput of the system.



Plot 6 represents a comparison of greedy and non-greedy algorithms for the two parameters: Total number of nodes and the Average Response time.



3 Topological Comparisons

For the testing of the three parameters: mesaage complexity, response time and throughput, we anazlye based on three topology: linear tree, star tree and binary tree to see every parameters difference. For accuracy, we run each topology for 5times, and average it parameters. Relevant files for this section: test.csv, graph.R.

Linear tree topology:
Nodes: 20
Message Complexity: 3386.8
Response Time: 207.6
Throughput(Per Requests): 62.4

Linear tree topology(Greedy):
Nodes: 20
Message Complexity: 668.9
Response Time: 41
Throughput(Per Requests): 12.6

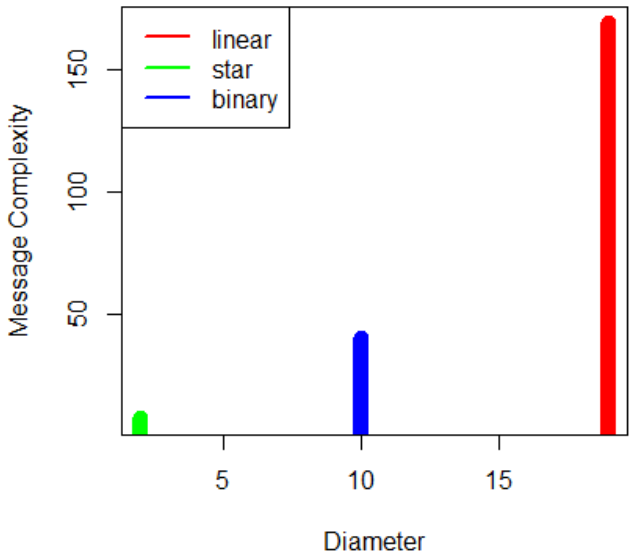
Star tree topology:
Nodes: 21
Message Complexity: 145.2
Response Time: 730.2
Throughput(Per Requests): 12.6

Star tree topology(Greedy):
Nodes: 21
Message Complexity: 146.8
Response Time: 735
Throughput(Per Requests): 11.8

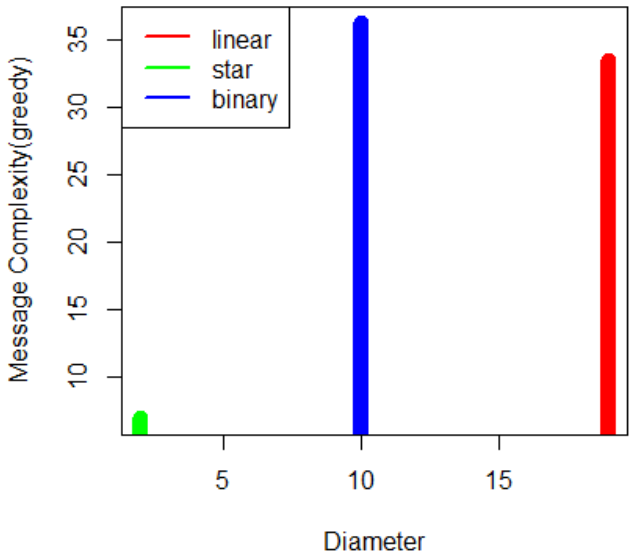
Binary tree topology:
Nodes: 31
Message Complexity: 1249.2
Response Time: 524.8
Throughput(Per Requests): 37.6

Binary tree topology(Greedy):
Nodes: 31
Message Complexity: 1121.6
Response Time: 531
Throughput(Per Requests): 39.6

Message Complexity(Per Node):

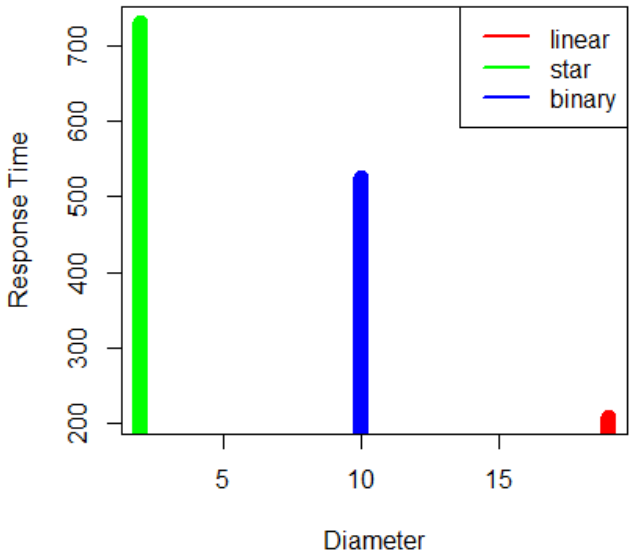


Message Complexity Greedy(Per Node):

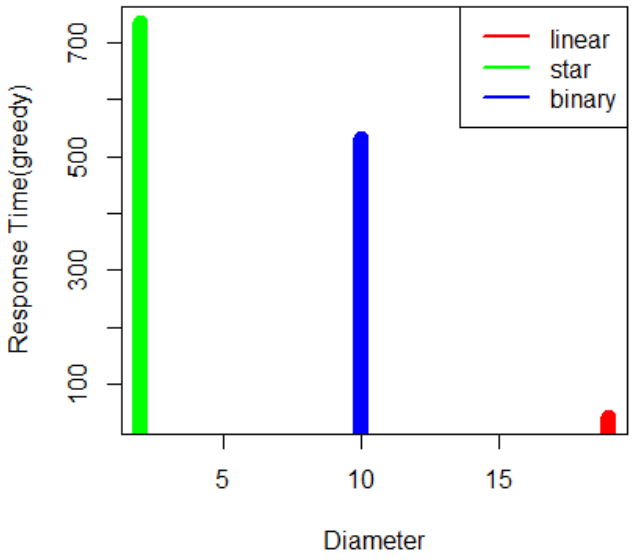


For the analysis of message complexity, we take the message per nodes to make it easier to analyze. For the message complexity, we can see that the greedys one is far less than the non-greedy one, especially for the linear tree topology.

Response Time:

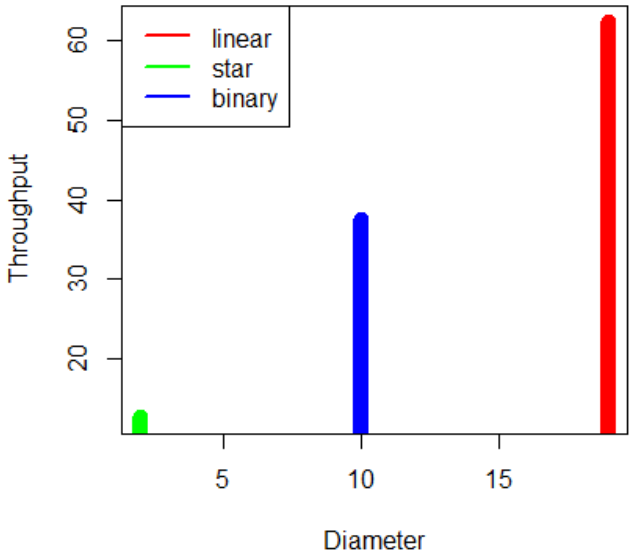


Response Time(Greedy)

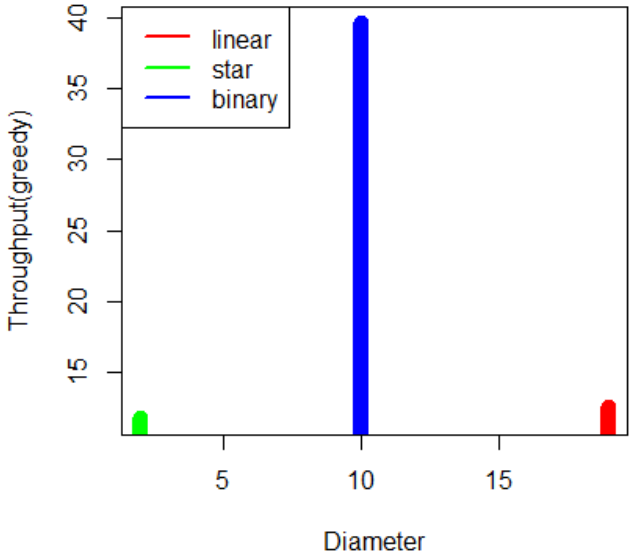


For the response time, even we implement the greedy method, the response time do not affect by it.

Throughput:



Throughput (Greedy):



For the throughput per each requests, we can observe that the greedy method is workable on the linear topology and star topology, but not the binary.