



# Implementation of Custom Routing Algorithm in Cloud

By  
Disha Bhattacharya  
Biswajeet Chakraborty  
Palash Dey  
Ananya Laha

# Basic Idea



- ❖ Algorithm for distributing data to all nodes in a network based on the concept of percolation centrality (PC) or betweenness centrality (BC)
- ❖ Enhancement of the controlled flooding algorithm
  - Adding concept of percolation centrality

# Purpose



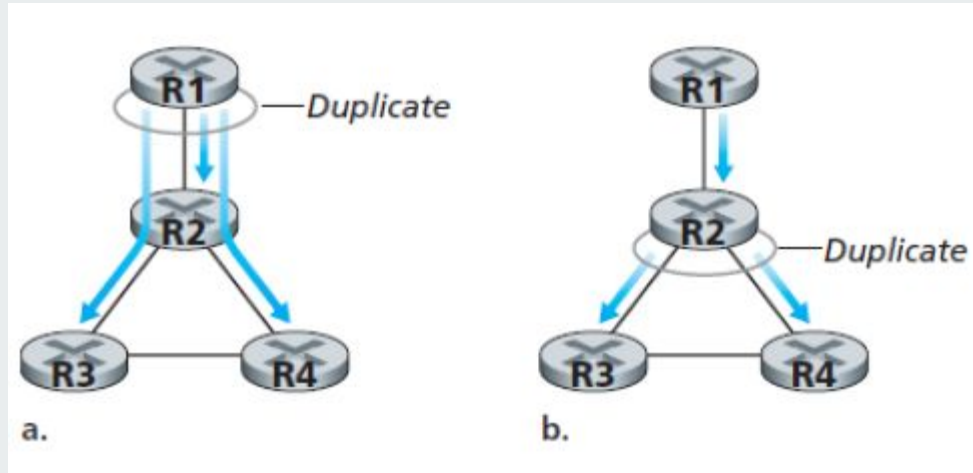
- ❖ The purpose of the routing algorithm is to make decisions for the router concerning the optimal paths for data distribution
- ❖ The router uses the routing algorithm to get the path that would best serve to transport the data throughout the network
- ❖ The routing algorithm that our protocol uses is a major factor in the performance of our routing environment

# Broadcasting



- ❖ Message is destined to all network devices
- ❖ Most straightforward way: N-way-unicast
- ❖ Broadcast Algorithms:
  - 1. Uncontrolled Flooding
  - 2. Controlled Flooding
  - 3. Spanning Tree Broadcast

# Broadcasting



Source duplication vs in-network duplication

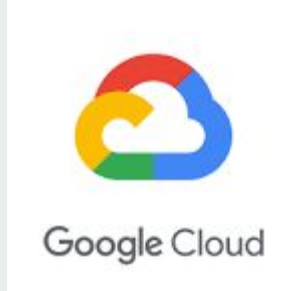
# Brief Description



- ❖ Similar to the controlled flooding algorithm
- ❖ Enhance the flooding algorithm using the concept of percolation centrality
- ❖ Send a message that percolates via the nodes of the network
- ❖ Time taken will be the least by using the concept of percolation centrality

# Implementation

Tech Stack:



# Algorithm Implementation



## Algorithm 1

To start routing from node with highest Betweenness Centrality

1. procedure
2. graphPC = descending\_PercCentrality(G)
3. for  $i \leftarrow 0, n-1$  do
4.     graphPC[i].MARK = False
5. for  $i \leftarrow 0, n-1$  do
6.     Call enhanced\_flooding(graphPC[i])



# Algorithm Implementation

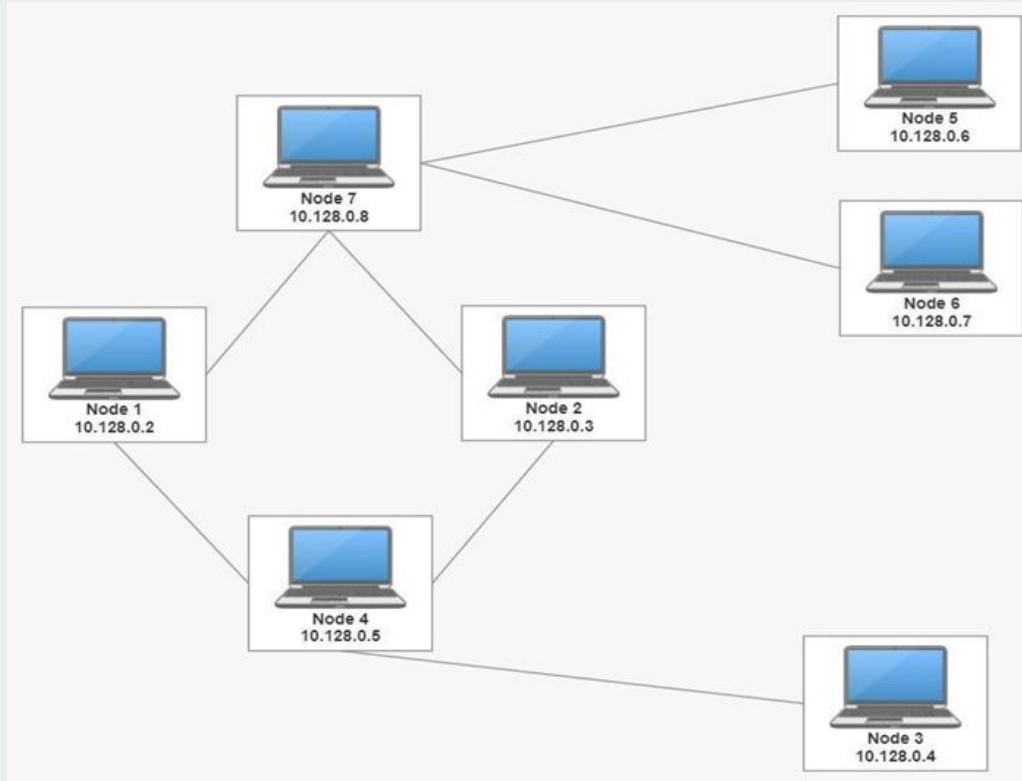


## Algorithm 2

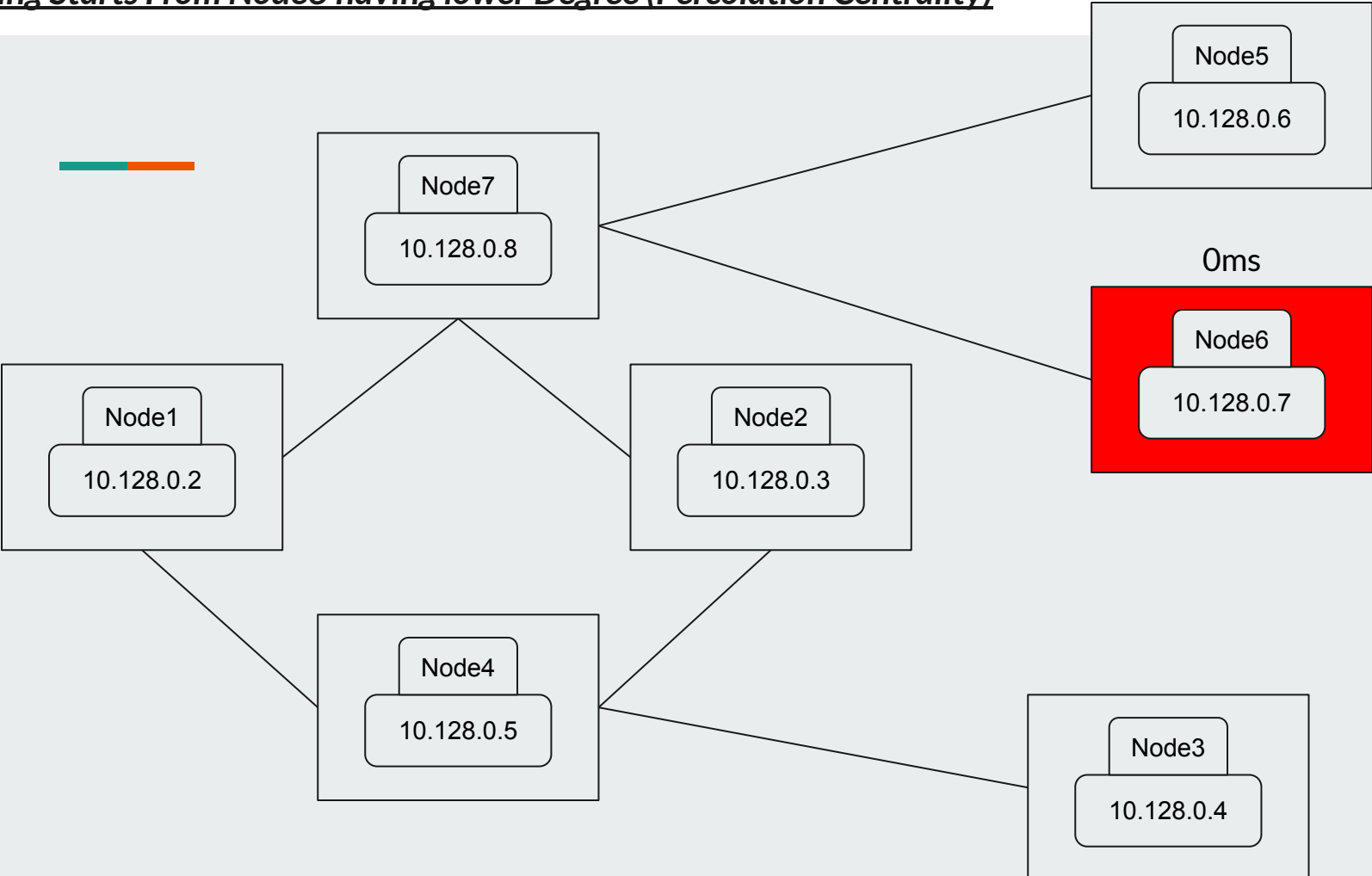
Algorithm for controlled flooding mechanism

1. procedure enhanced\_flooding(v)
2.   if v.MARK = False then
3.     v.MARK = True
4.     Accept message in v
5.     parfor each node k E v.adjacent() do
6.         Call enhanced\_flooding(k)
7.     end parfor

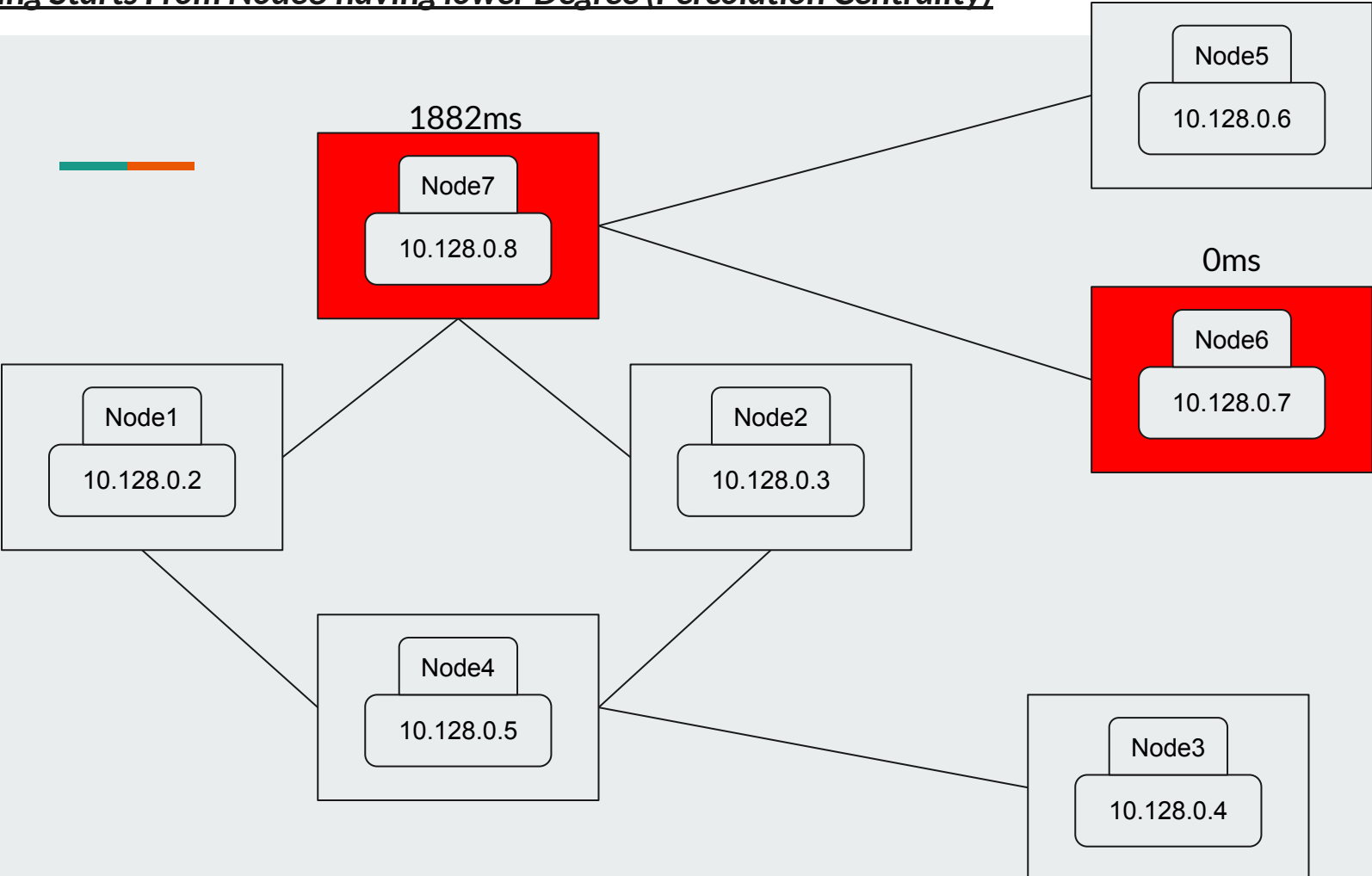
# Network Formation



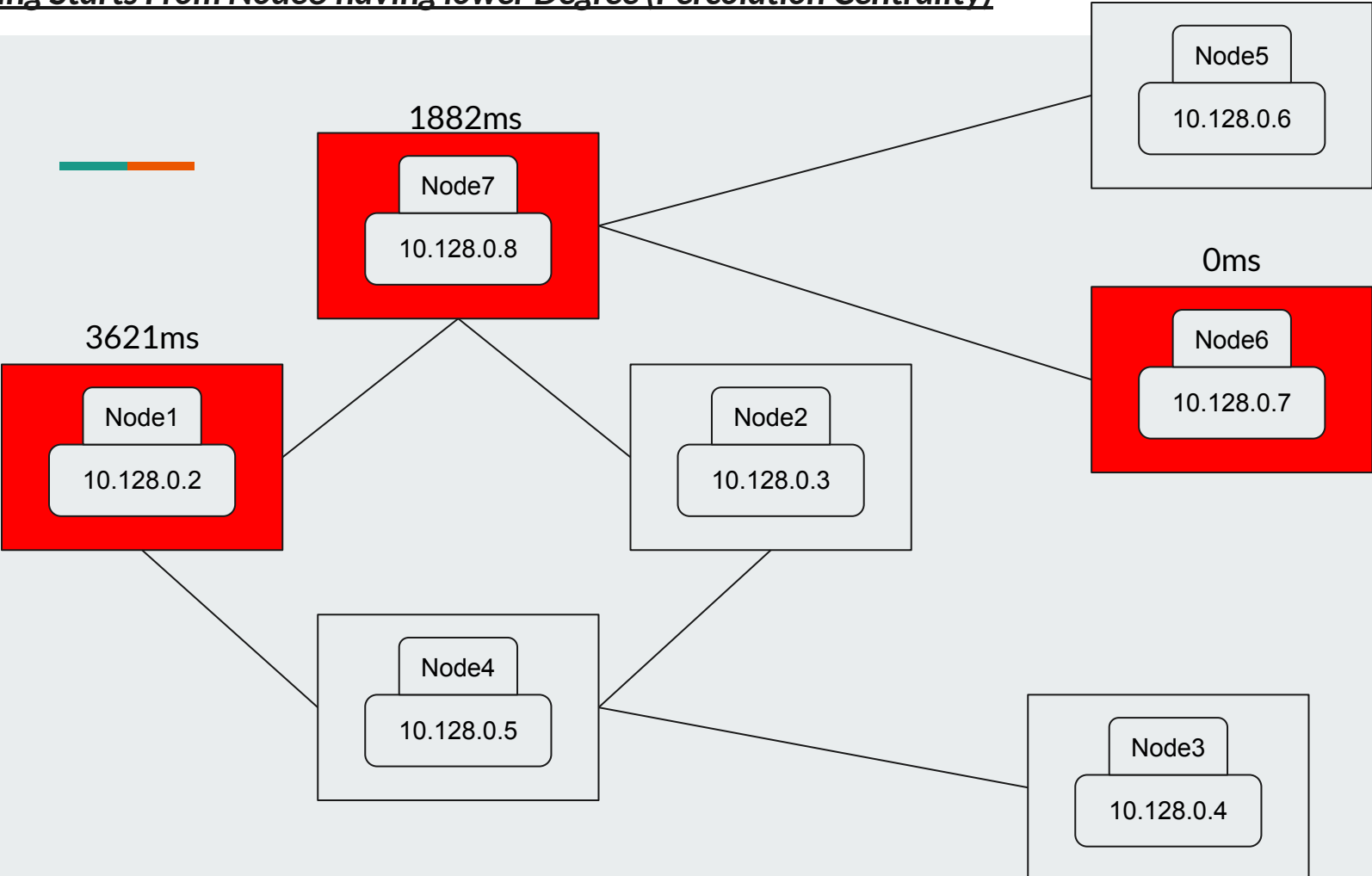
**Routing Starts From Node6 having lower Degree (Percolation Centrality)**



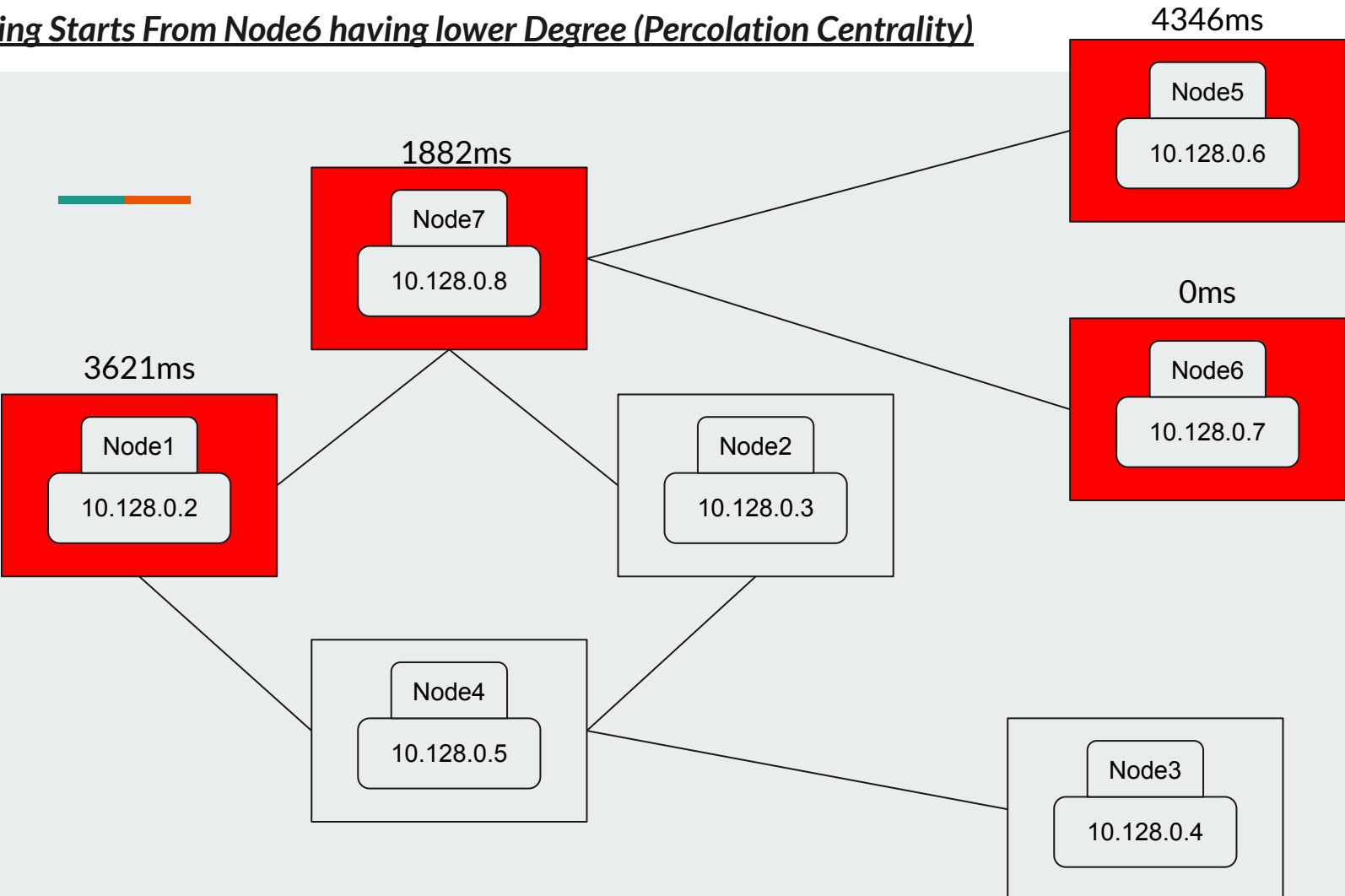
**Routing Starts From Node6 having lower Degree (Percolation Centrality)**



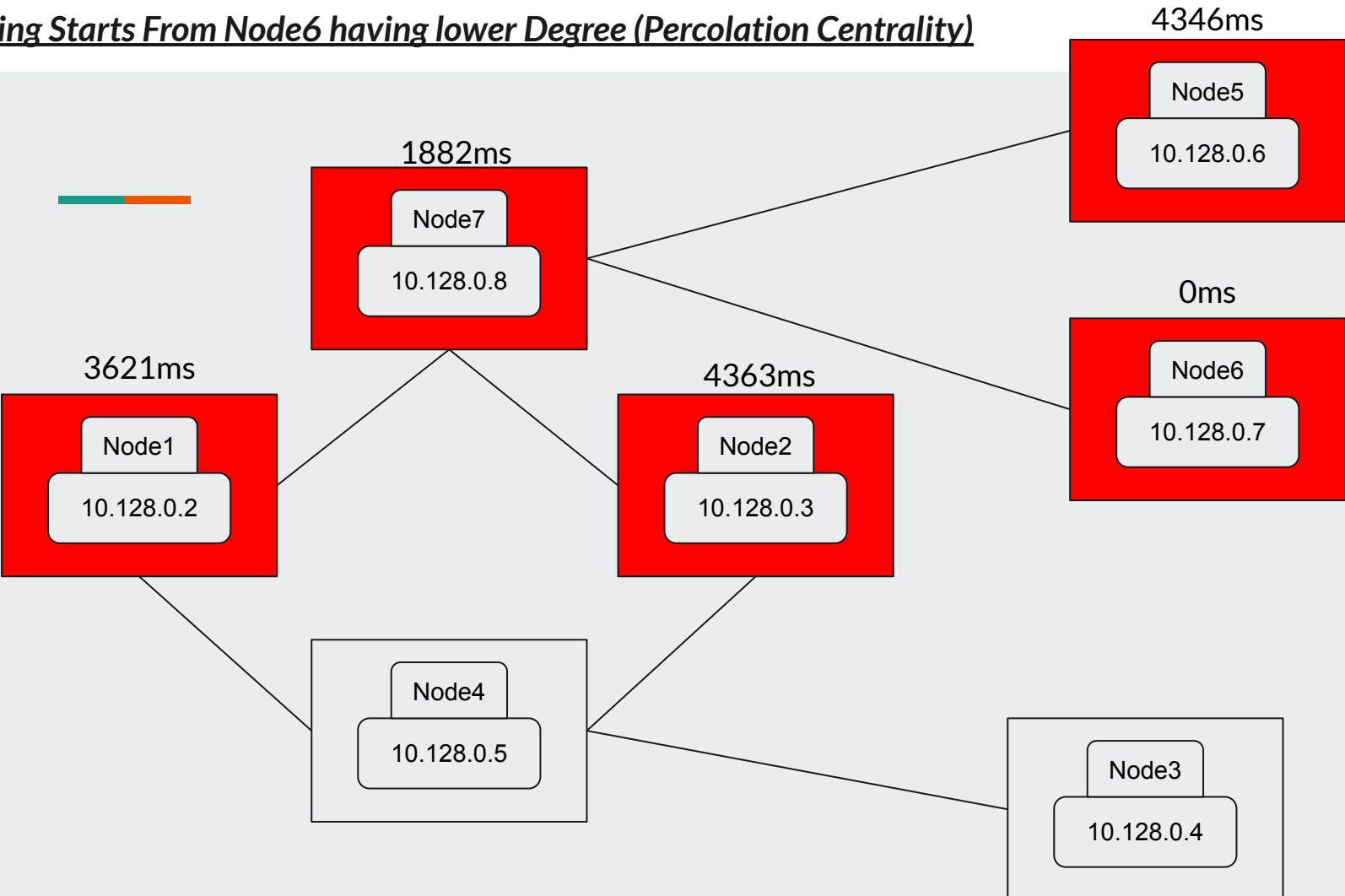
**Routing Starts From Node6 having lower Degree (Percolation Centrality)**



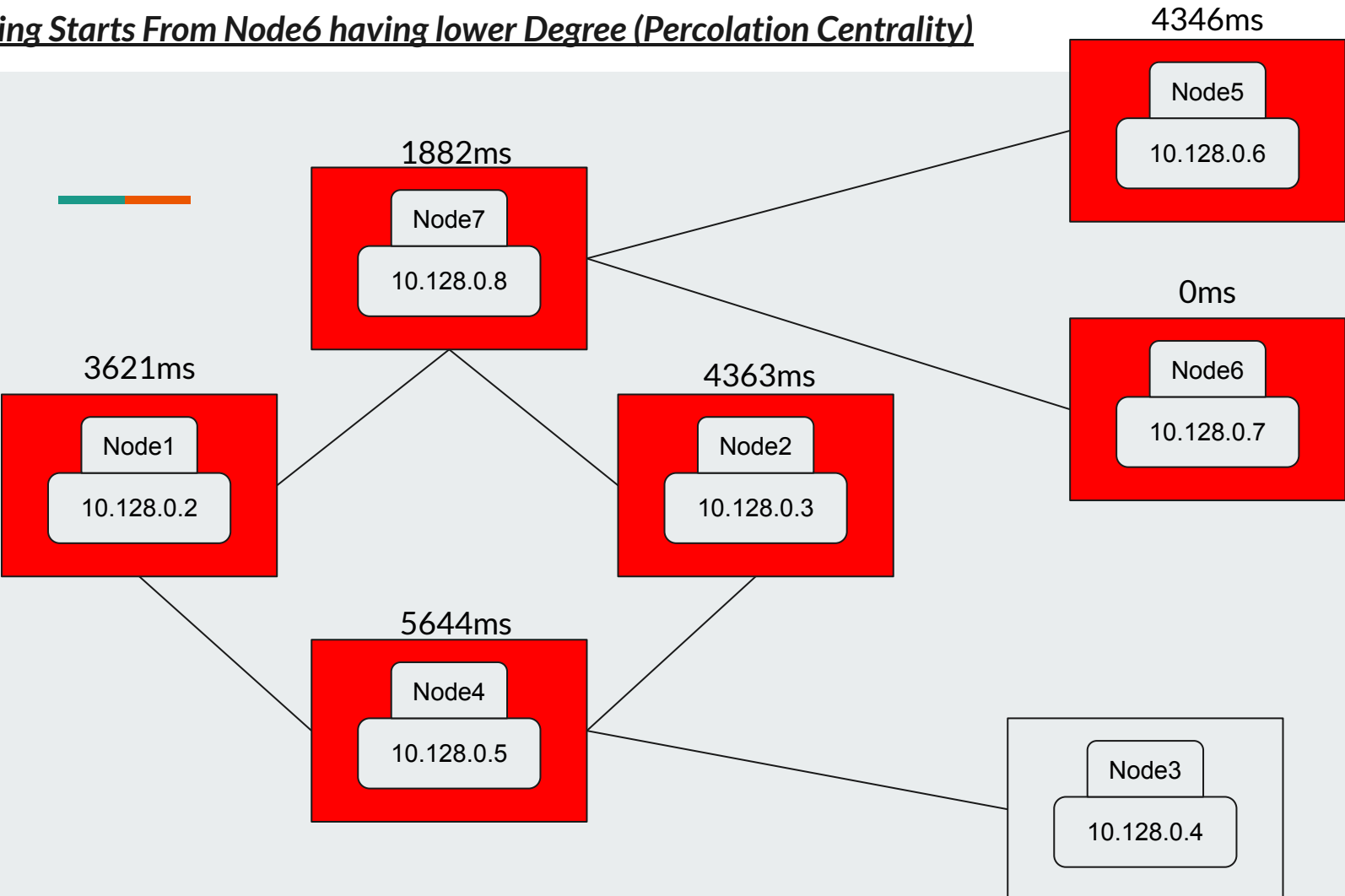
**Routing Starts From Node6 having lower Degree (Percolation Centrality)**



**Routing Starts From Node6 having lower Degree (Percolation Centrality)**

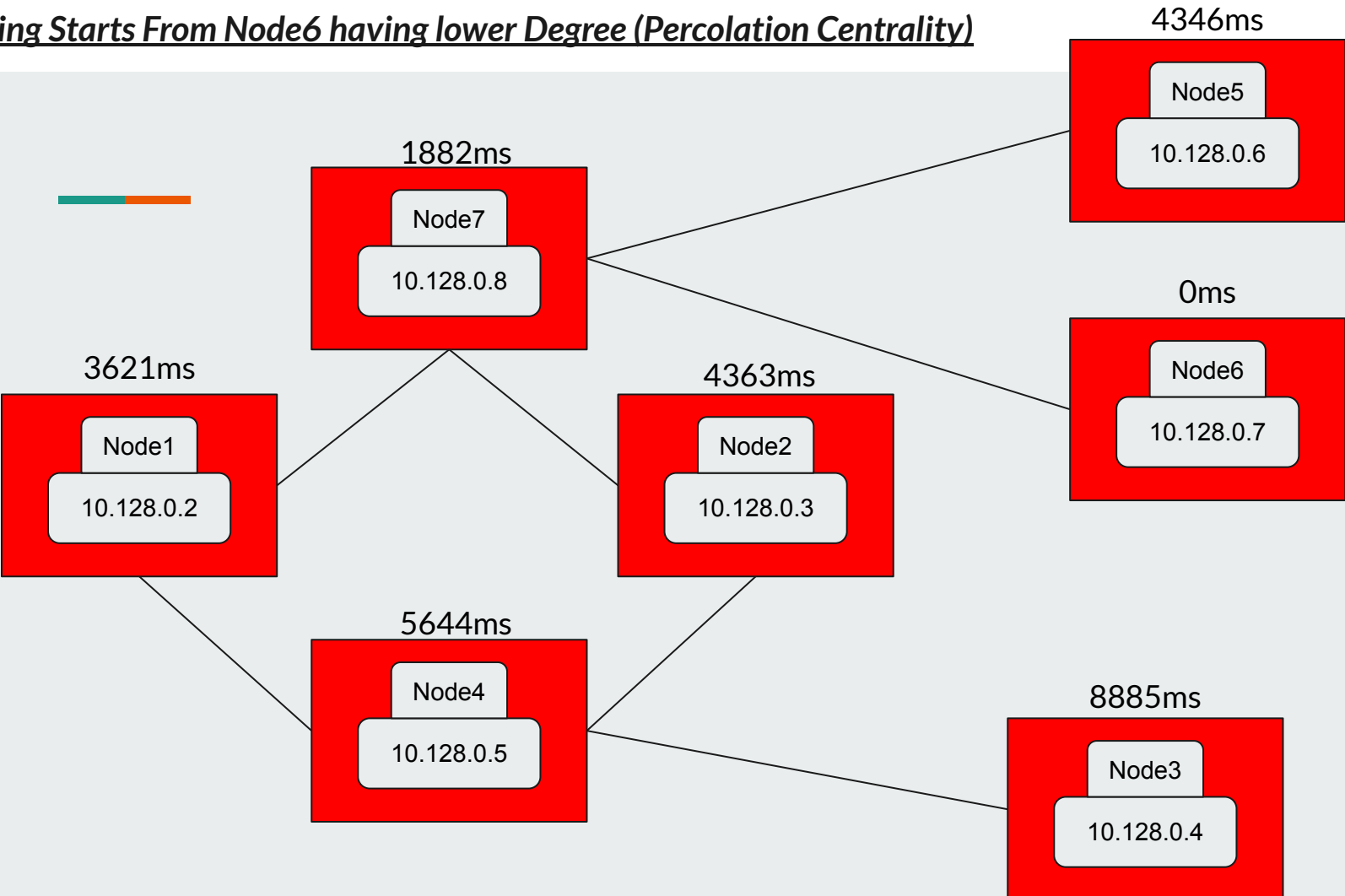


**Routing Starts From Node6 having lower Degree (Percolation Centrality)**

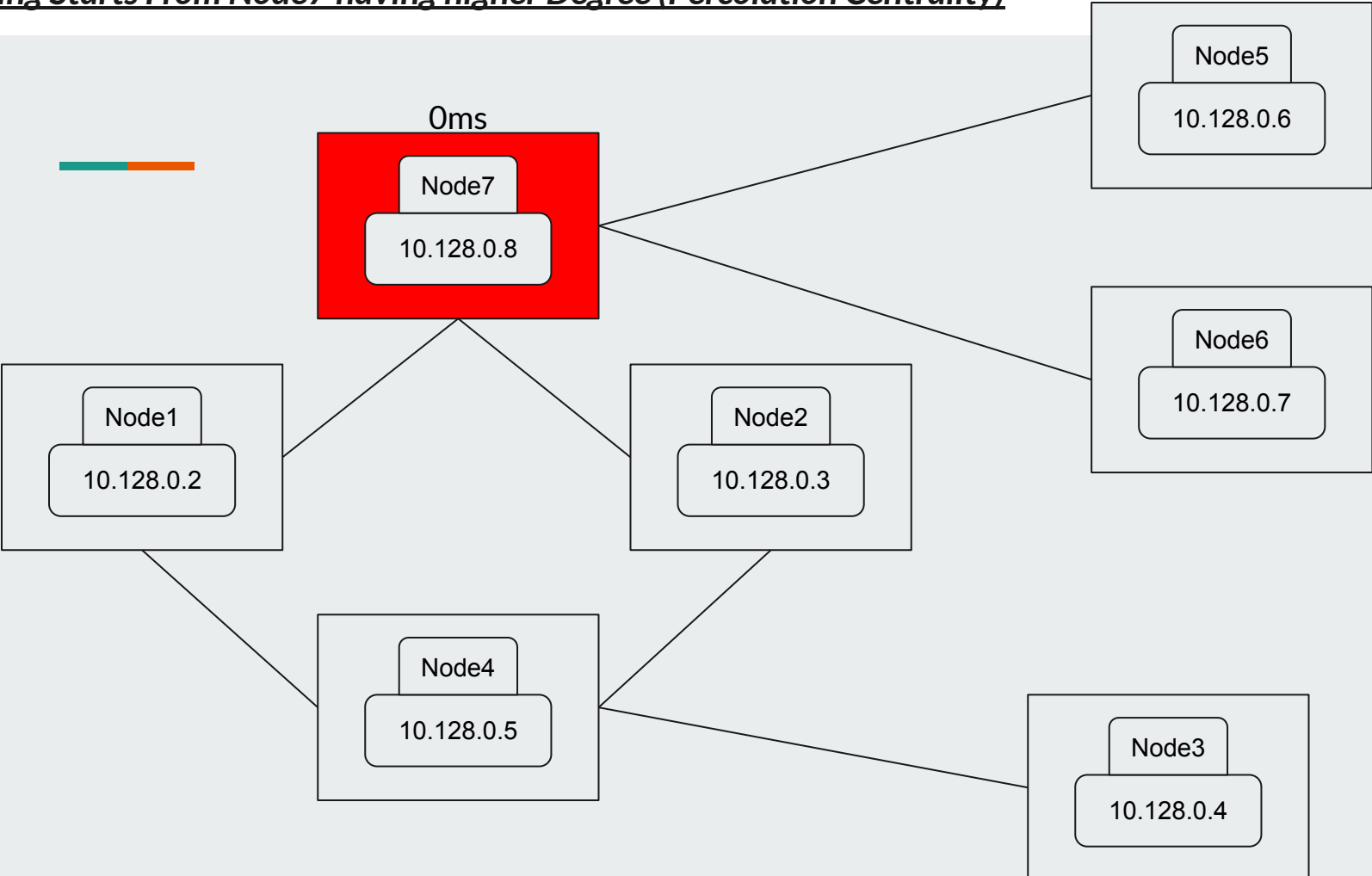




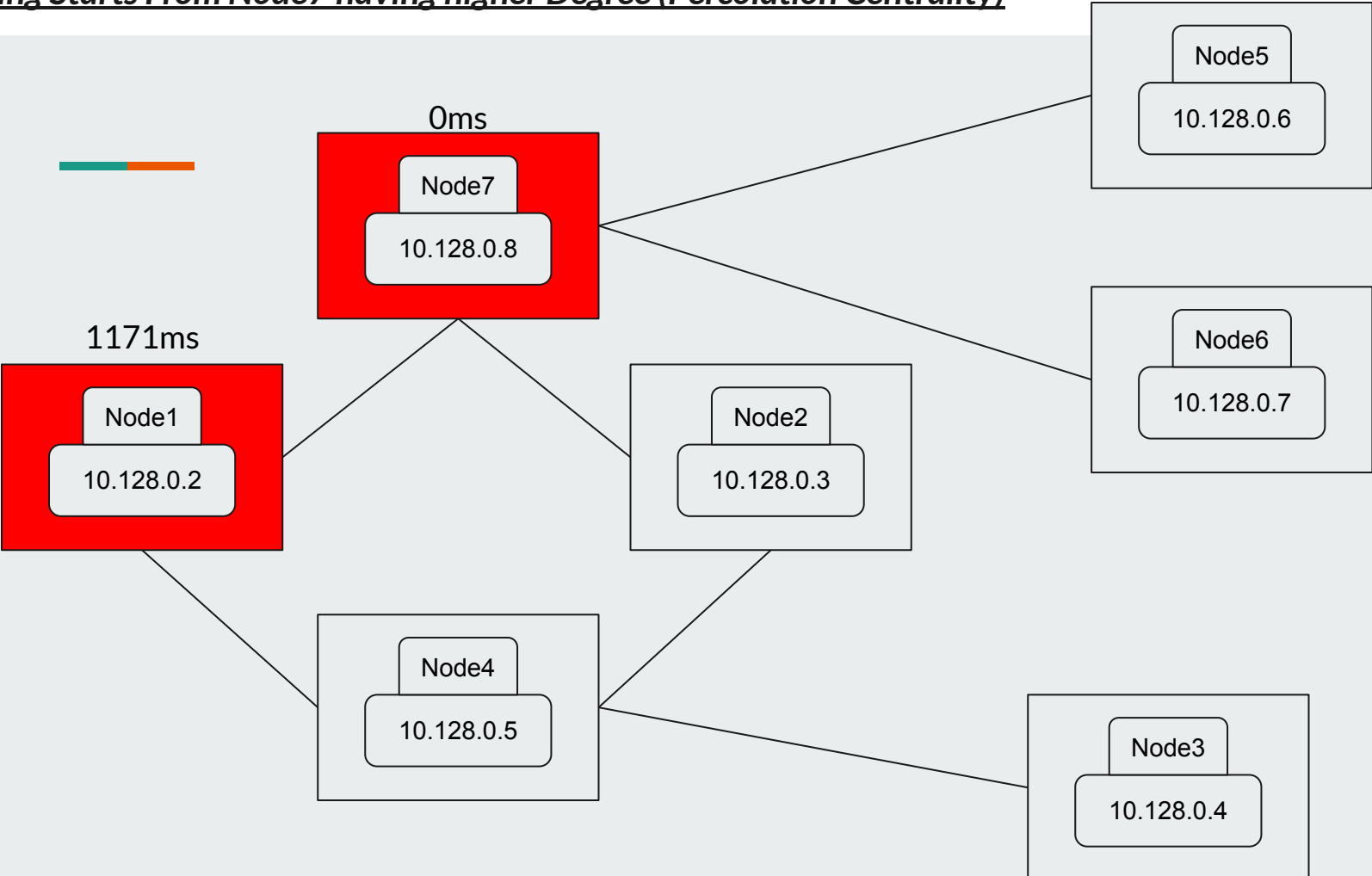
**Routing Starts From Node6 having lower Degree (Percolation Centrality)**



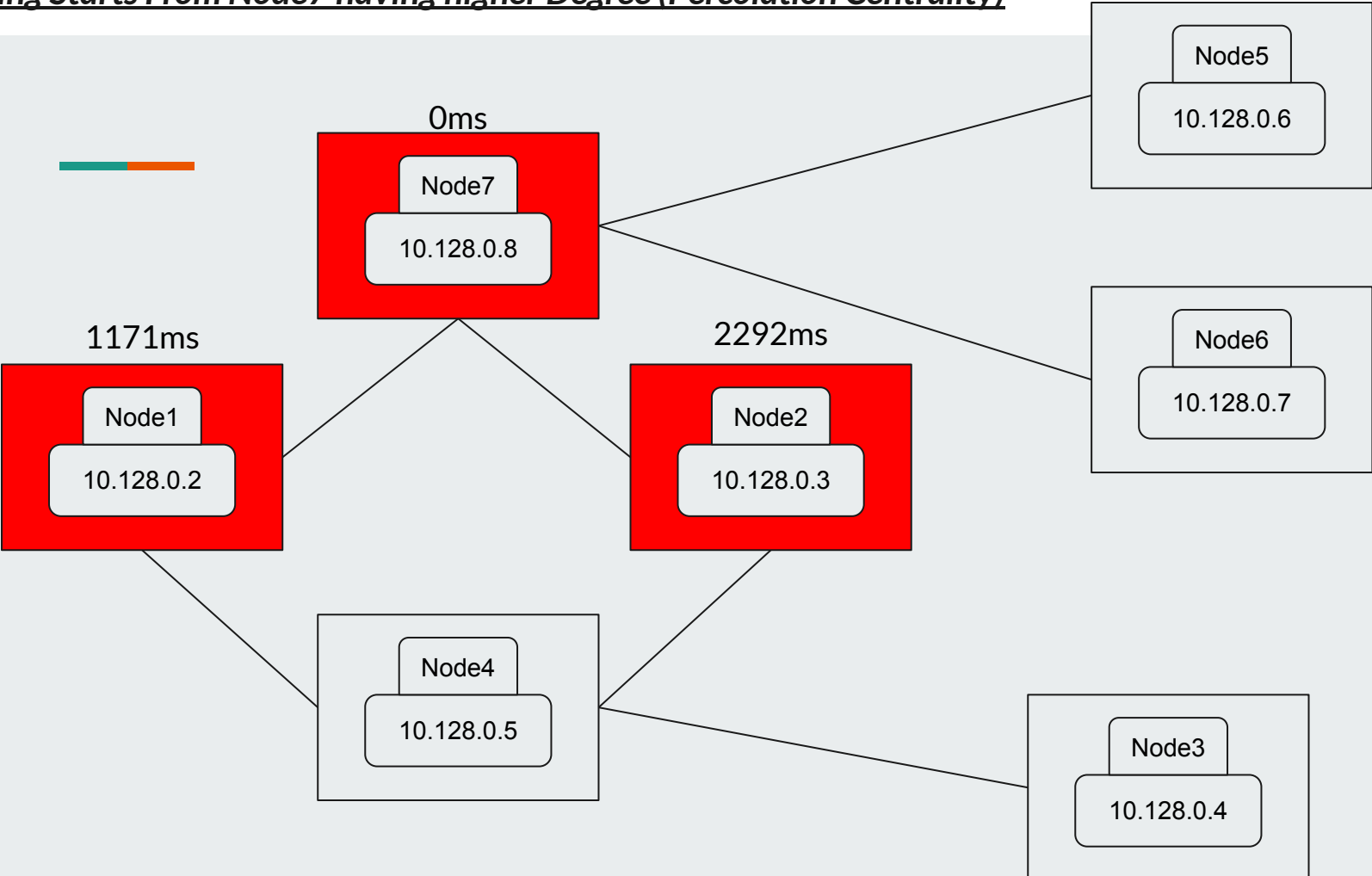
**Routing Starts From Node7 having higher Degree (Percolation Centrality)**



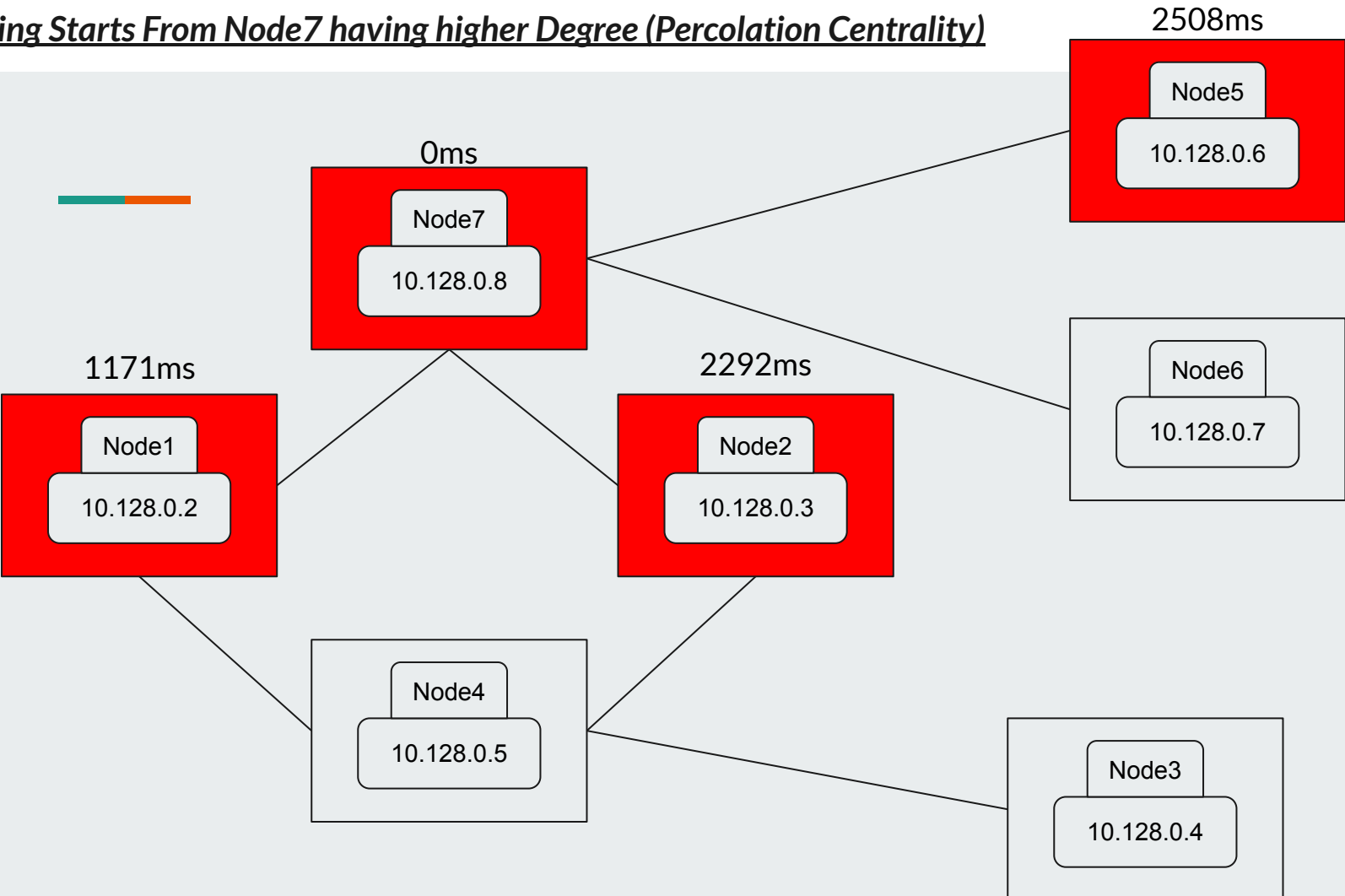
**Routing Starts From Node7 having higher Degree (Percolation Centrality)**



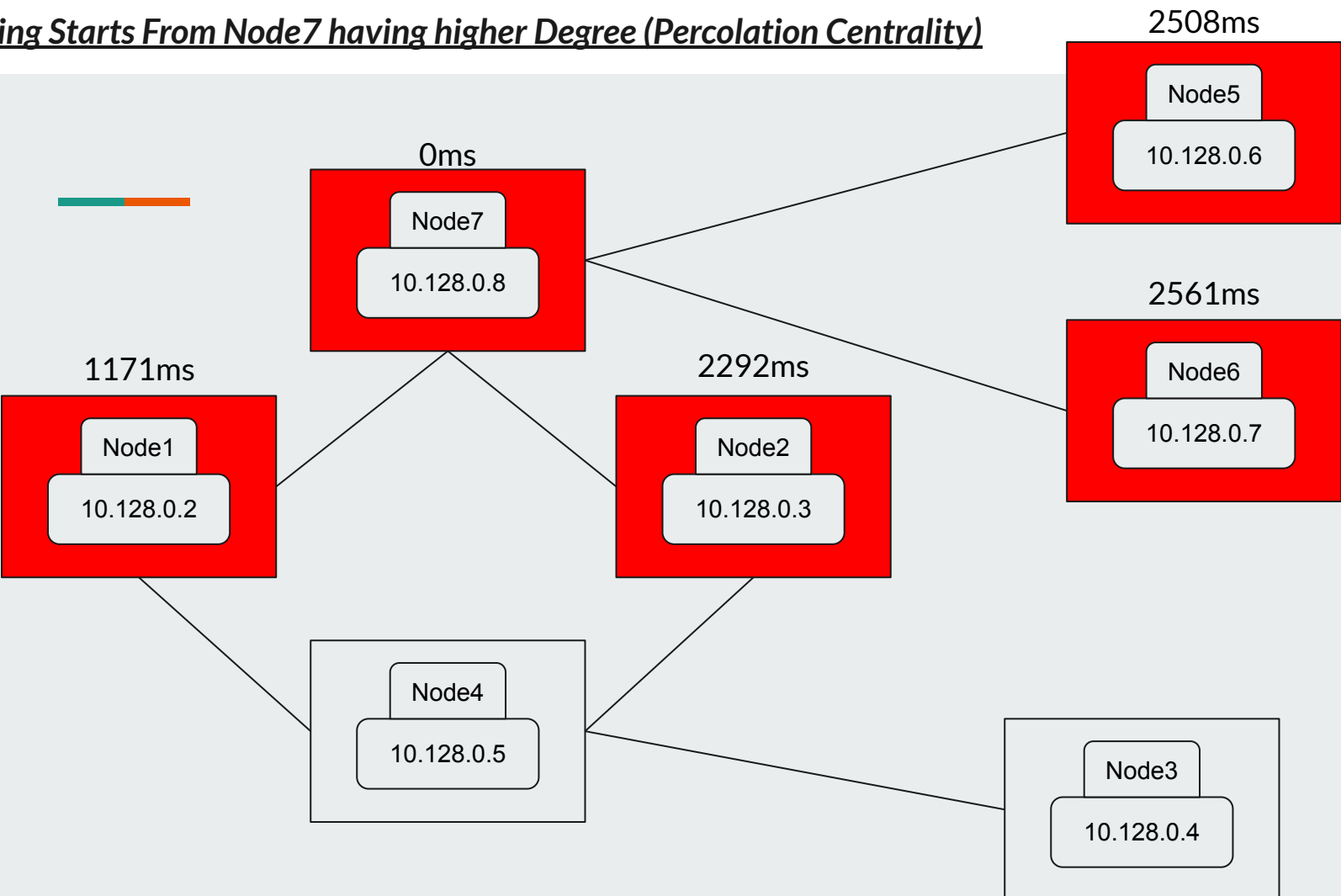
**Routing Starts From Node7 having higher Degree (Percolation Centrality)**



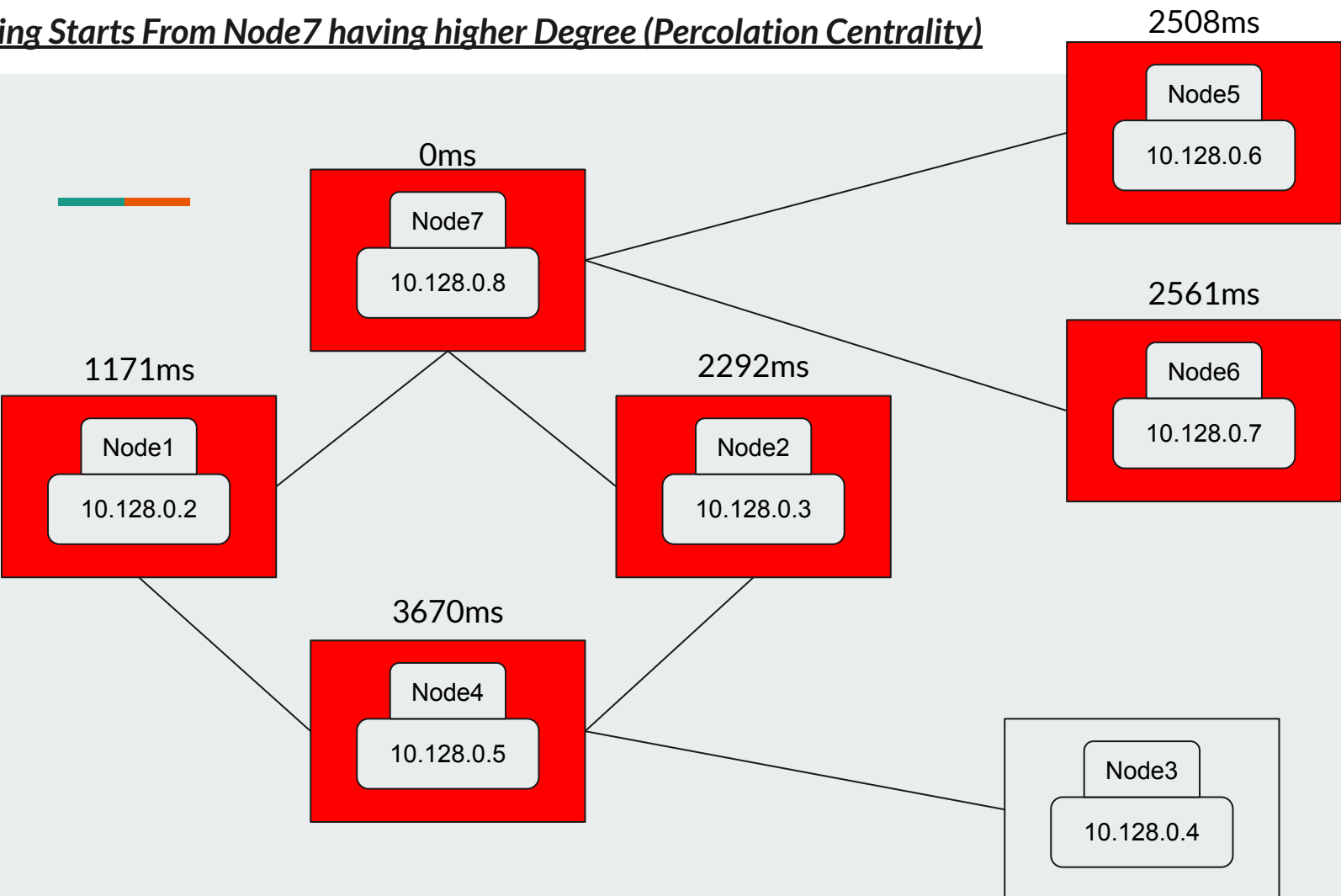
**Routing Starts From Node7 having higher Degree (Percolation Centrality)**



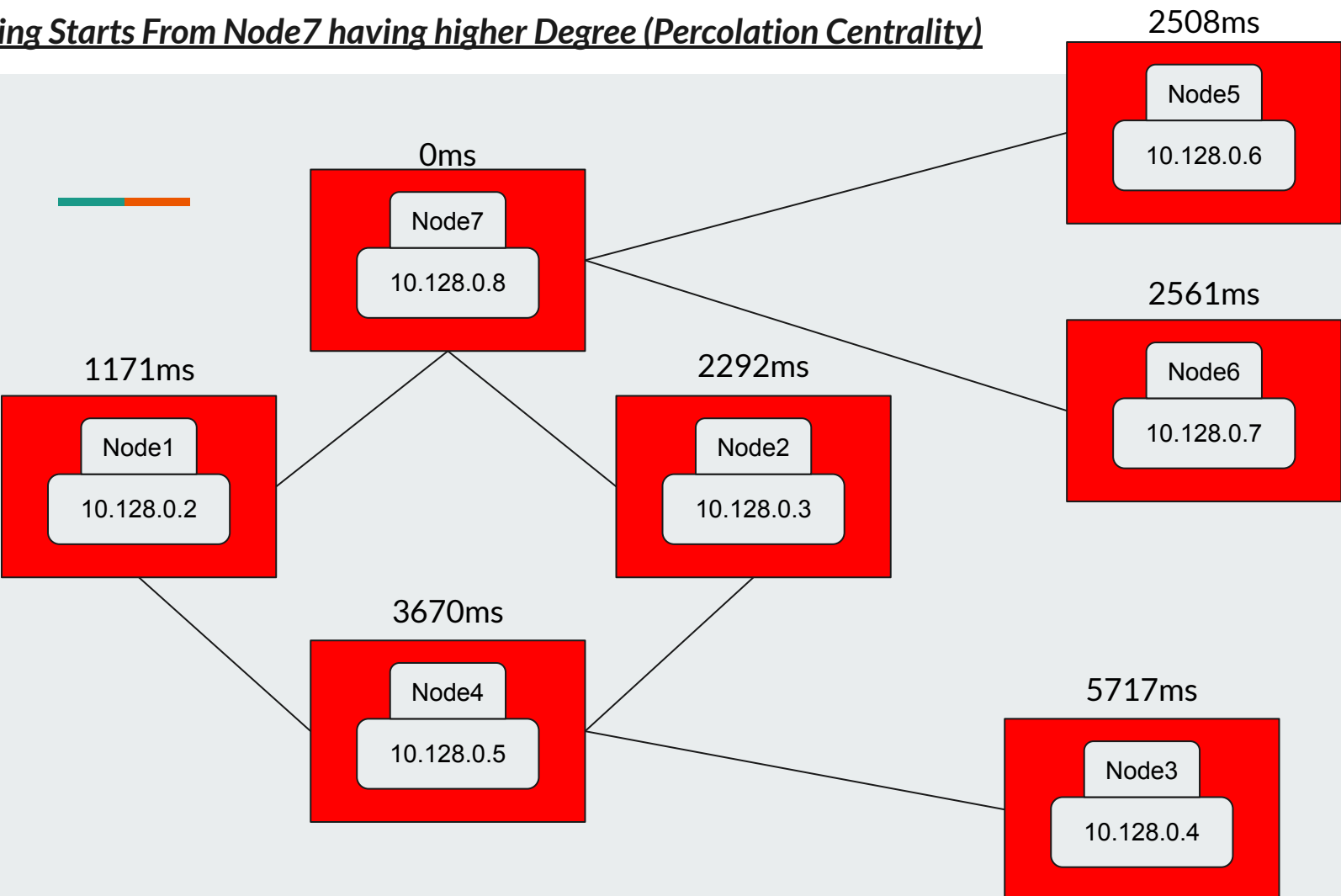
**Routing Starts From Node7 having higher Degree (Percolation Centrality)**



**Routing Starts From Node7 having higher Degree (Percolation Centrality)**



**Routing Starts From Node7 having higher Degree (Percolation Centrality)**



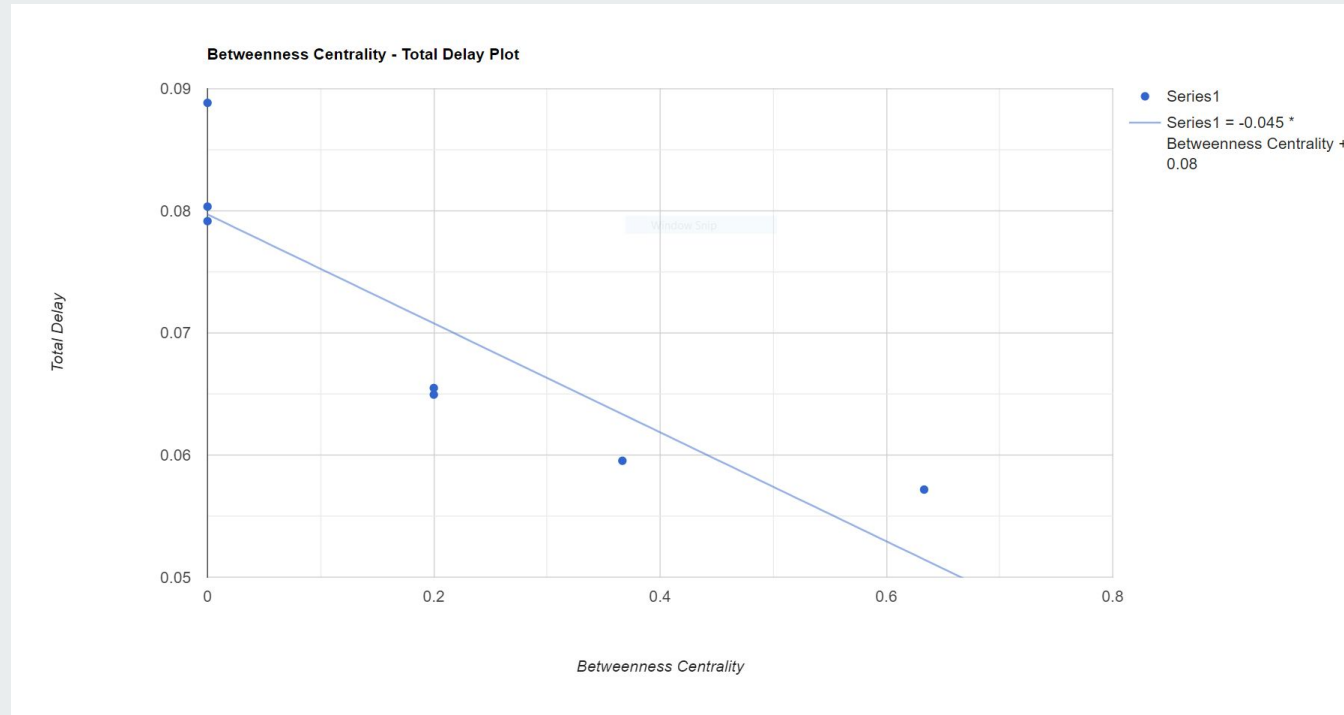


# Observation

Time taken for distribution:

Starting Node ↓	Time in msec →							Total Time
	1	2	3	4	5	6	7	
1	0	0.03729	0.03861	0.01765	0.05247	0.06495	0.02108	0.06495
2	0.03818	0	0.03931	0.0181	0.05266	0.06549	0.022	0.06549
3	0.03671	0.04063	0	0.01897	0.07665	0.07916	0.05724	0.07916
4	0.01862	0.02246	0.02237	0	0.05953	0.0583	0.03896	0.05953
5	0.03961	0.04219	0.08035	0.05755	0	0.04908	0.02033	0.08035
6	0.03621	0.04363	0.08885	0.05644	0.04346	0	0.01882	0.08885
7	0.01711	0.02292	0.05717	0.0367	0.02508	0.02561	0	0.05717

# Betweenness Centrality vs. Total Delay



# Conclusion



- ❖ It can be concluded from the scatter plot that the node having higher value of betweenness centrality will distribute the data/file in lesser time than the one having lower value of betweenness centrality.

# Future Scope



- ❖ Introduction of automation to minimize the number of manual steps to run the scripts for each routing.
- ❖ Distributing resources with complex format



***Thank You***