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 (Connectivity of nodes)

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.

- We check our algorithm by using Percolation Centrality, whether the highest PC node can deliver the packet, the fastest.

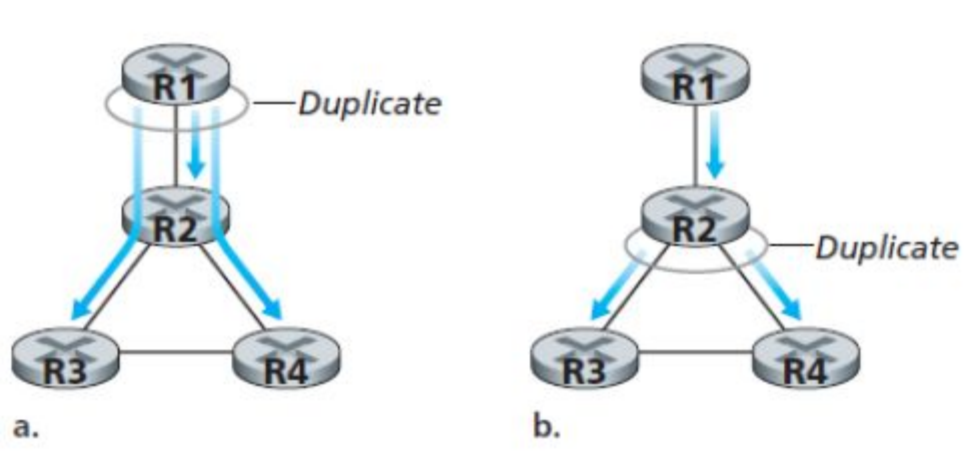
Broadcasting

- Message is destined to all network devices

- Most straightforward way: N-way-unicast : Needs no new network-layer routing protocol or forwarding functionality.

- 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:

Google Cloud

Algorithm Implementation

Algorithm 1

To start routing from node with highest Betweenness Centrality

```
procedure
graphPC = descending_PercCentrality(G)
for i <- 0, n-1 do
        graphPC[i].MARK = False
for i <- 0, n-1 do
        Call enhanced_flooding(graphPC[i])
```

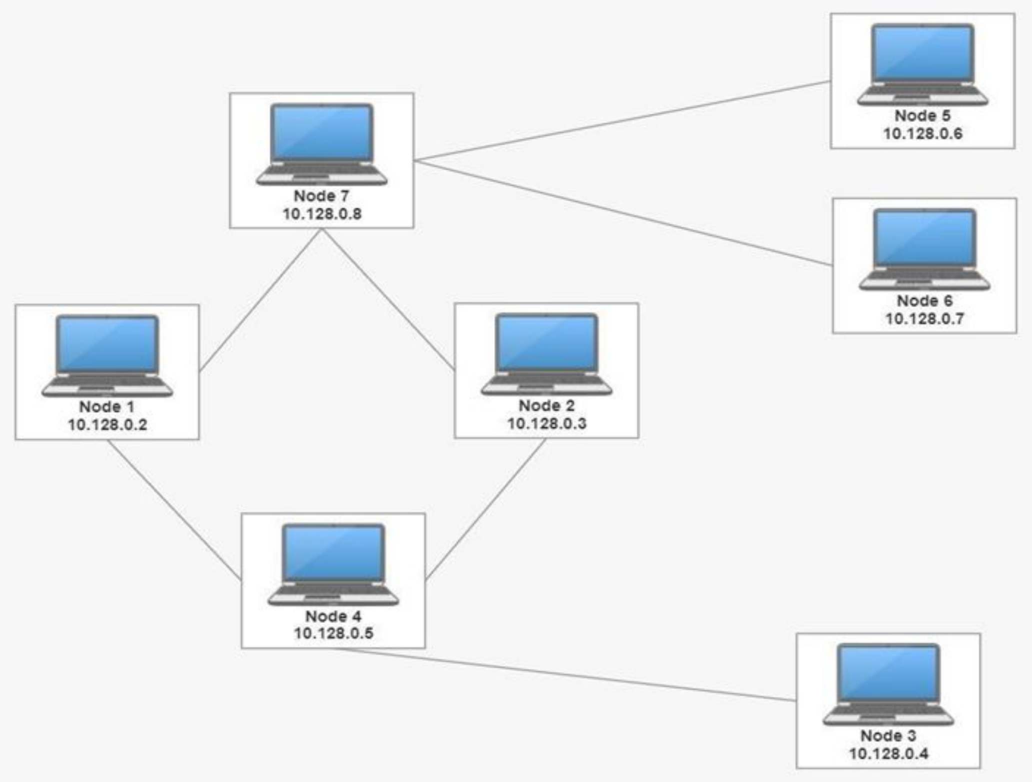
Algorithm Implementation

Algorithm 2

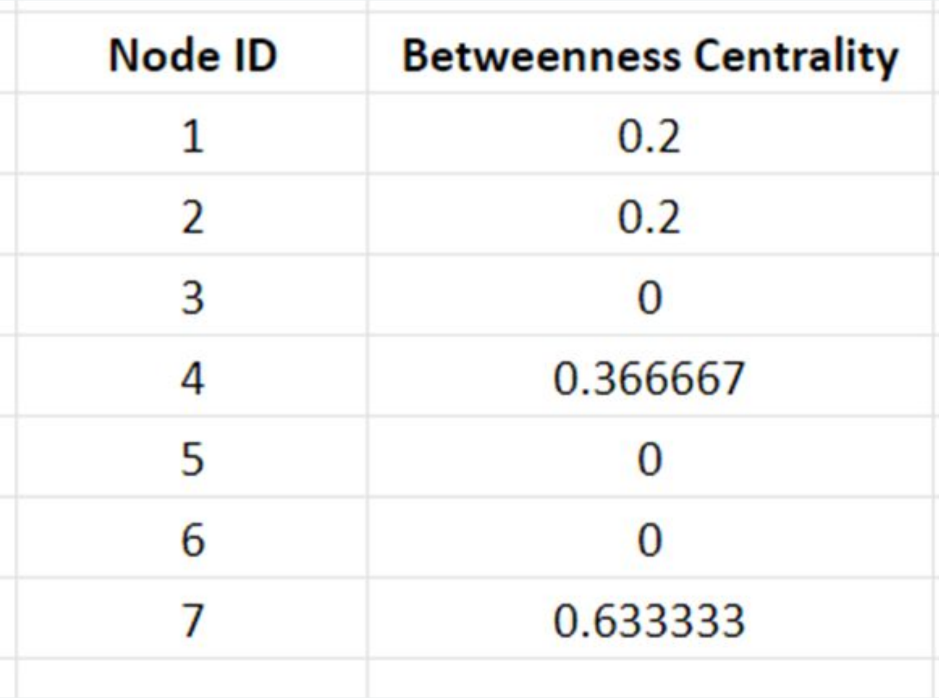
Algorithm for controlled flooding mechanism

```
procedure enhanced_flooding(v)
if v.MARK = False then
    v.MARK = True
    Accept message in v
    for each node k E v.adjacent() do
        Call enhanced_flooding(k)
    end for
```

Network Formation

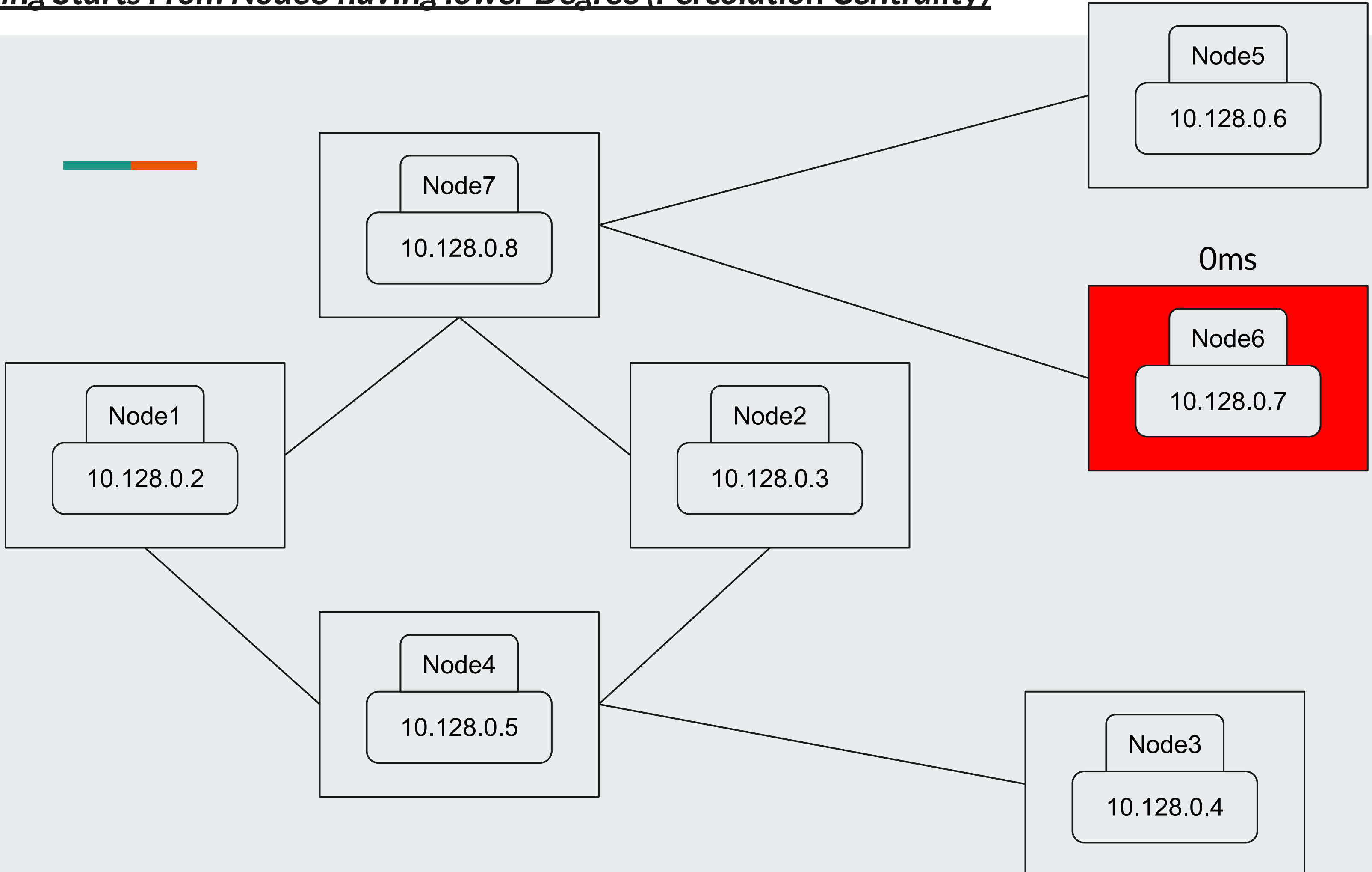


Betweenness Centrality of Nodes

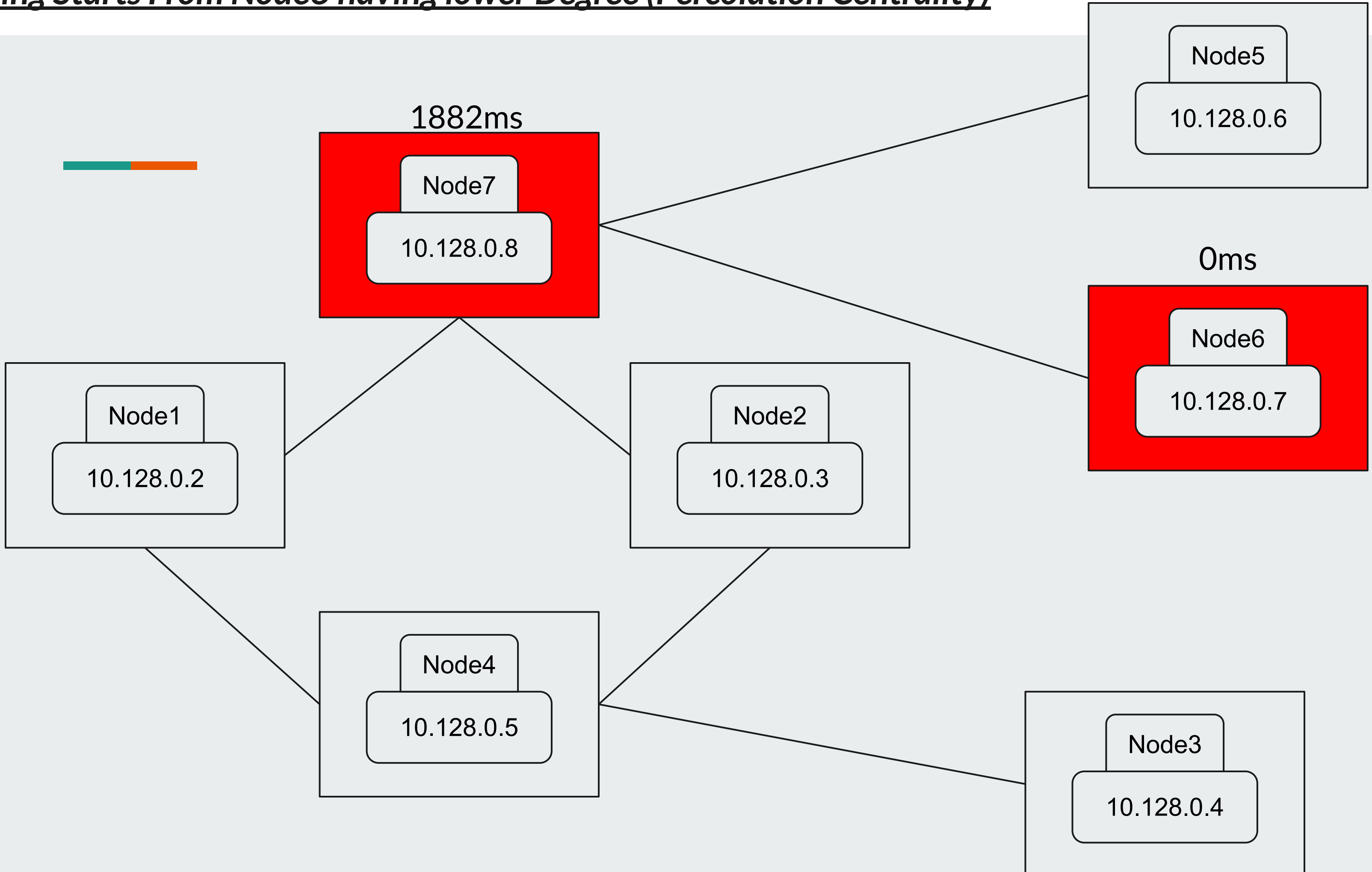


Node ID	Betweenness Centrality
1	0.2
2	0.2
3	0
4	0.366667
5	0
6	0
7	0.633333

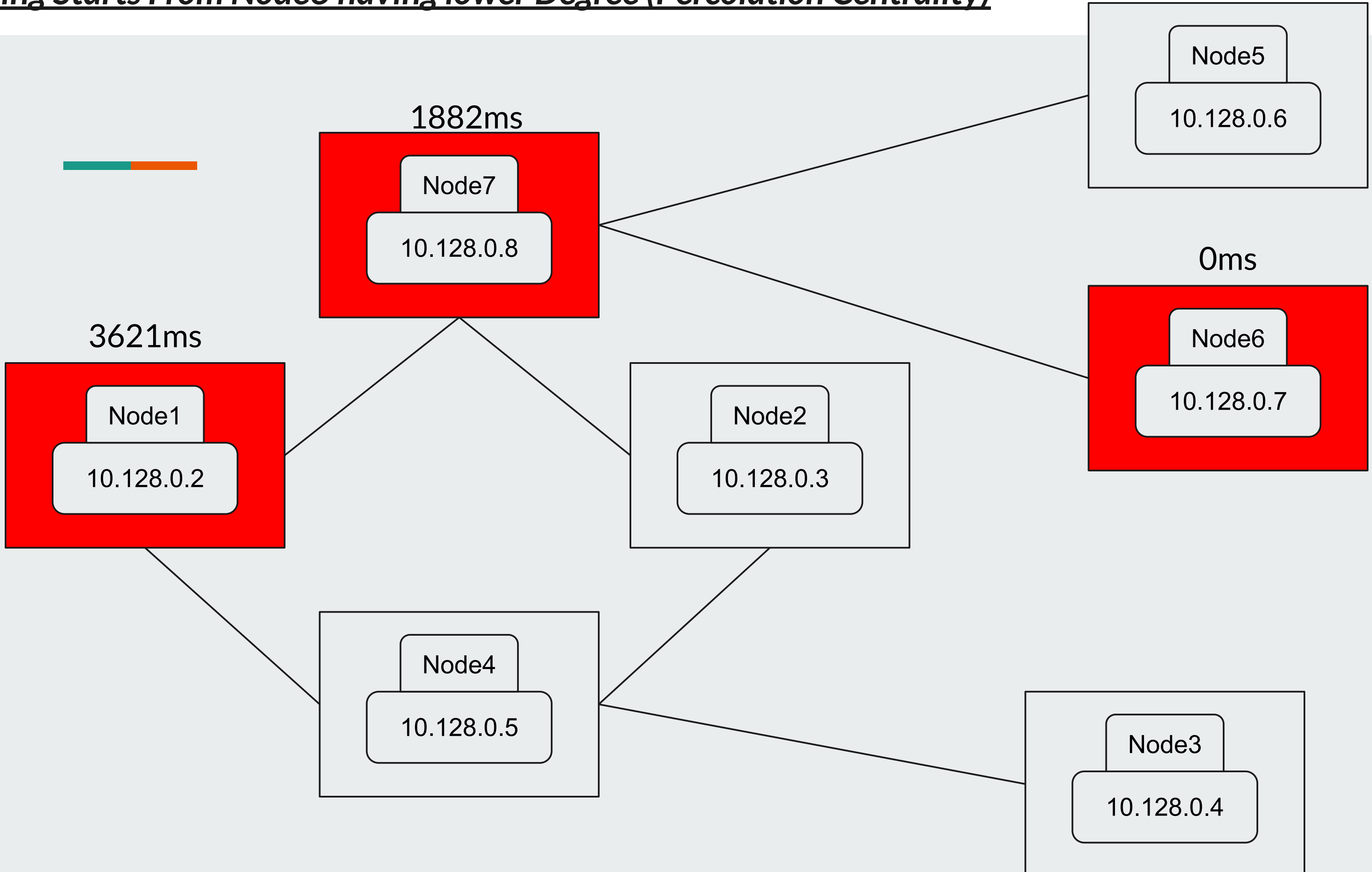
Routing Starts From Node6 having lower Degree (Percolation Centrality)



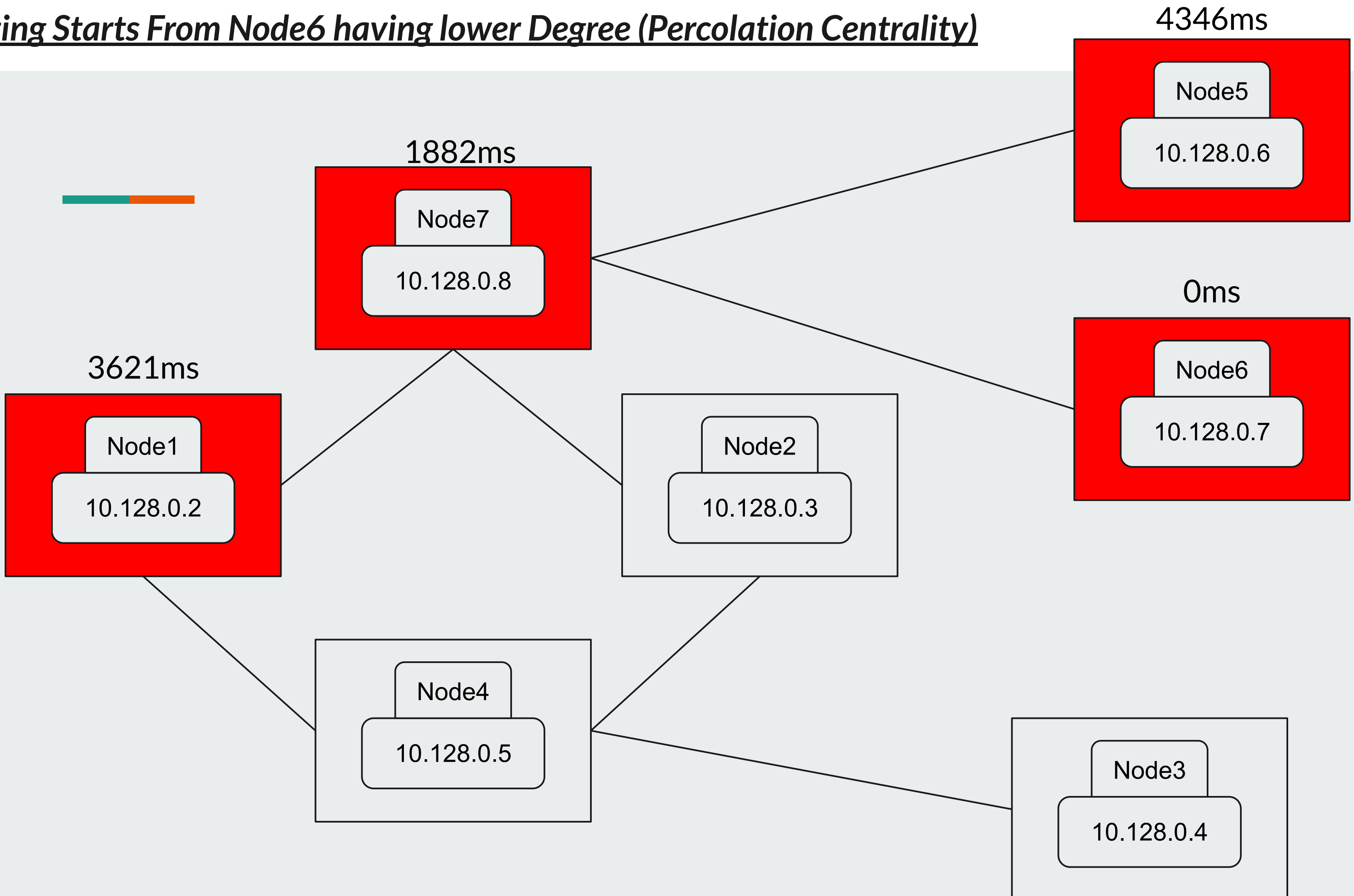
Routing Starts From Node6 having lower Degree (Percolation Centrality)



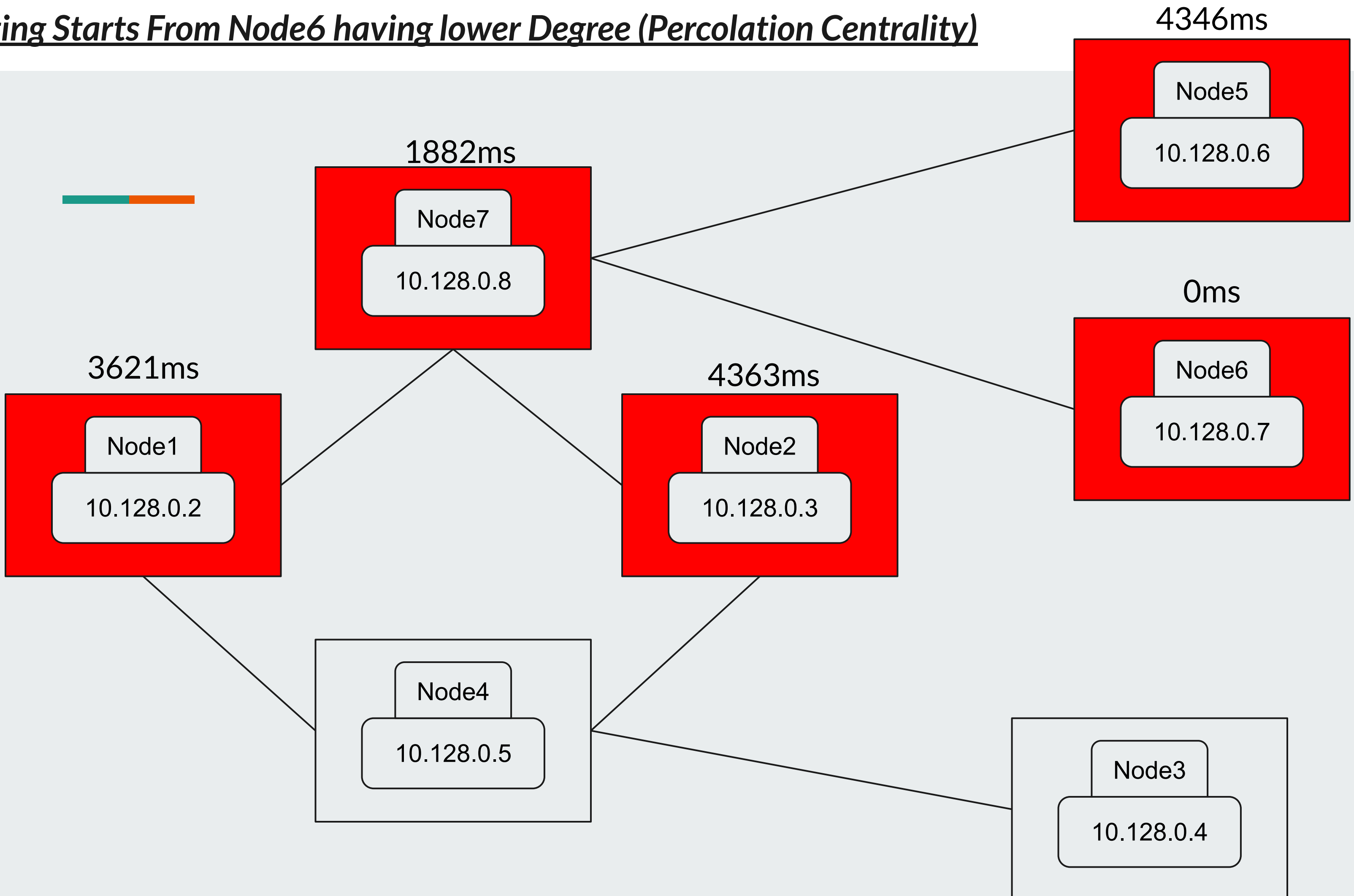
Routing Starts From Node6 having lower Degree (Percolation Centrality)



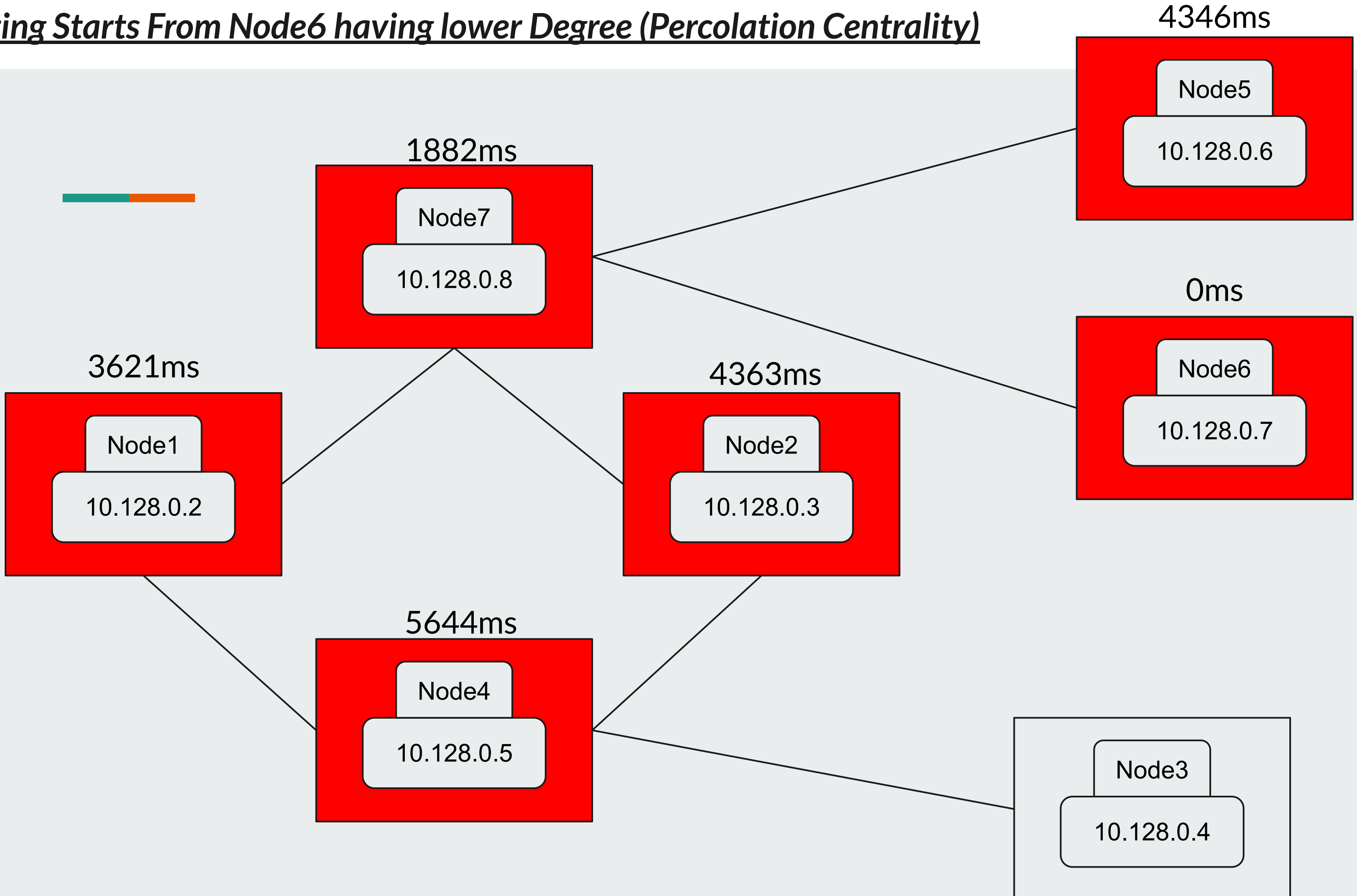
Routing Starts From Node6 having lower Degree (Percolation Centrality)



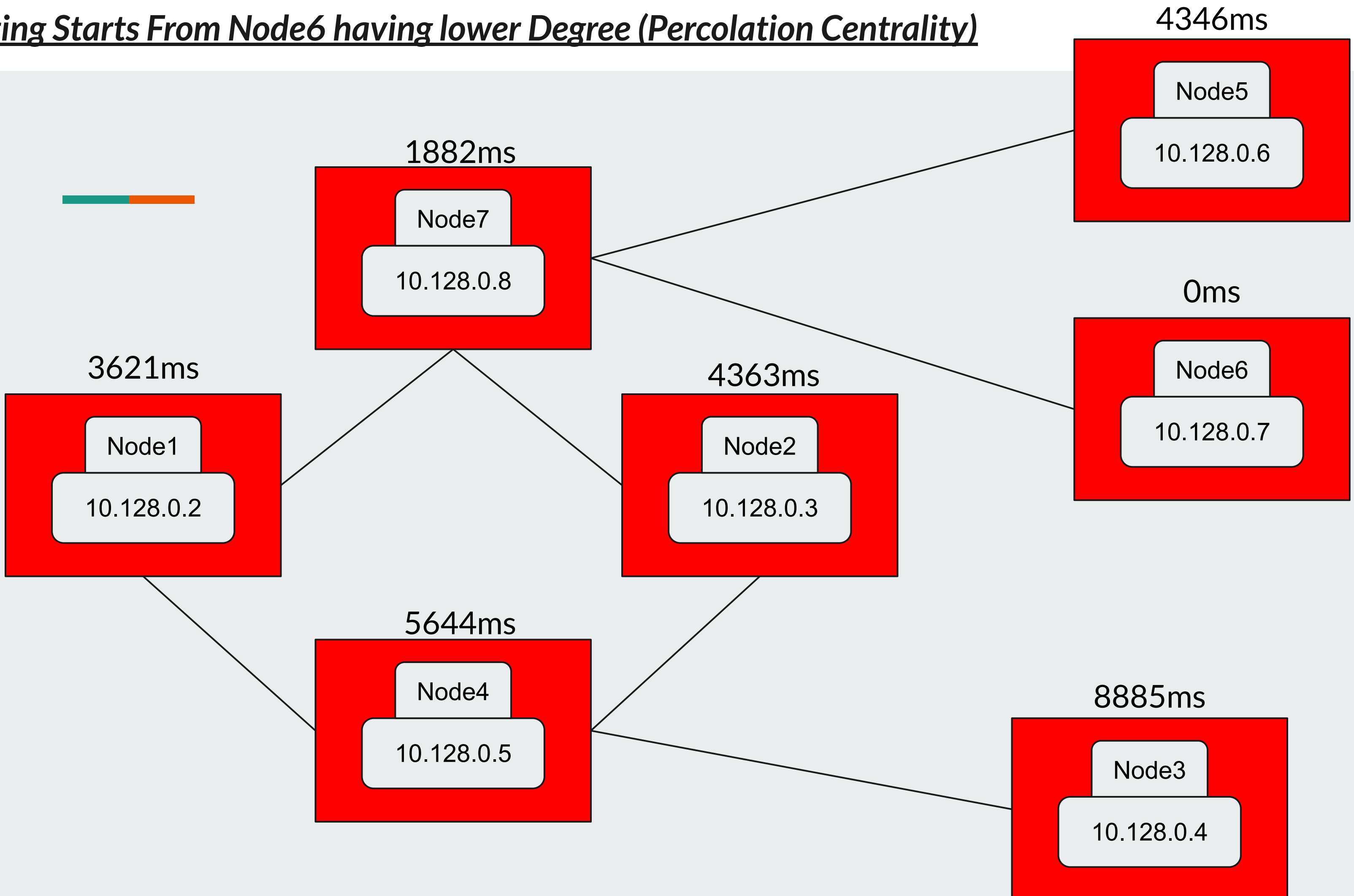
Routing Starts From Node6 having lower Degree (Percolation Centrality)



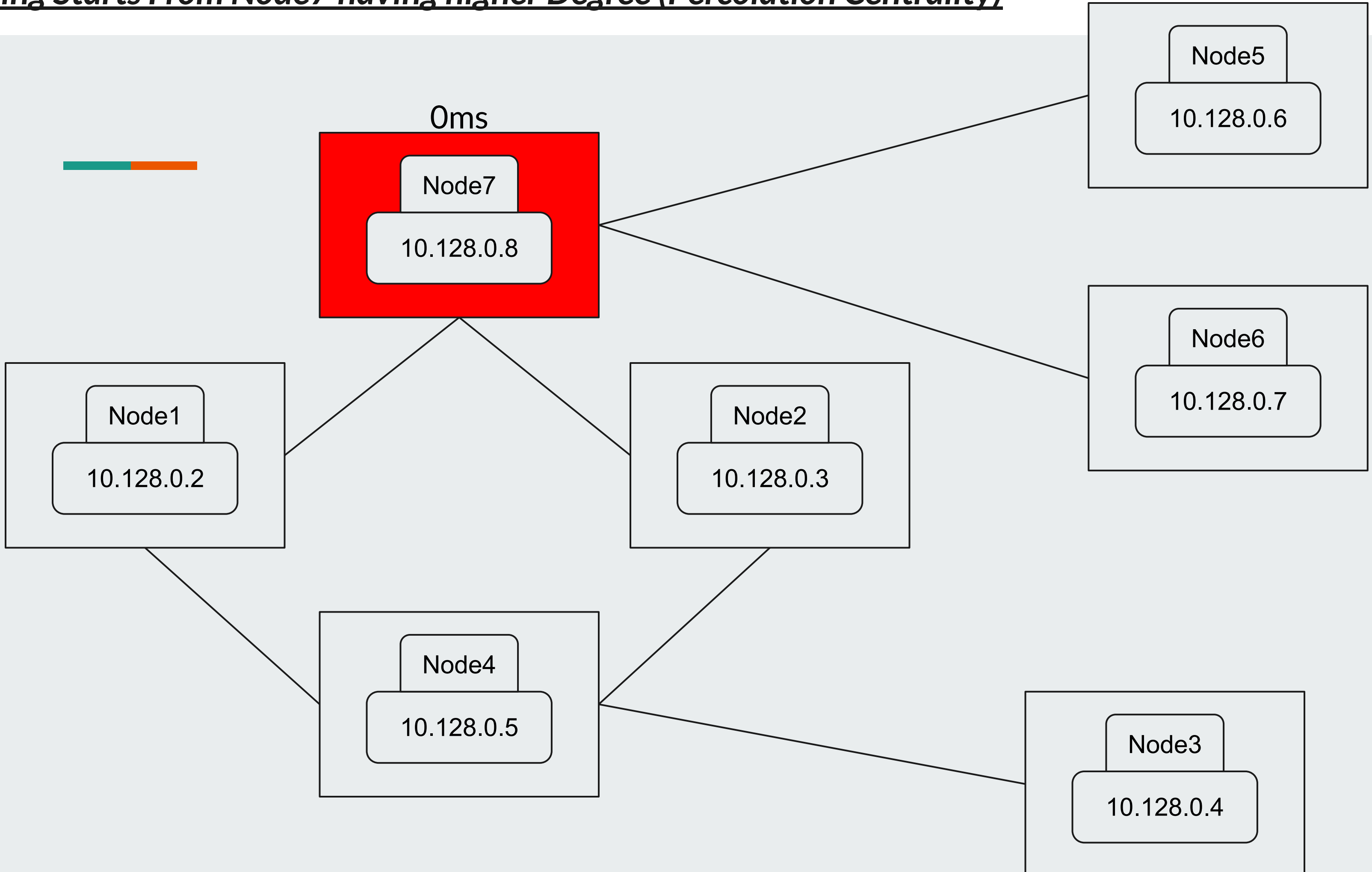
Routing Starts From Node6 having lower Degree (Percolation Centrality)



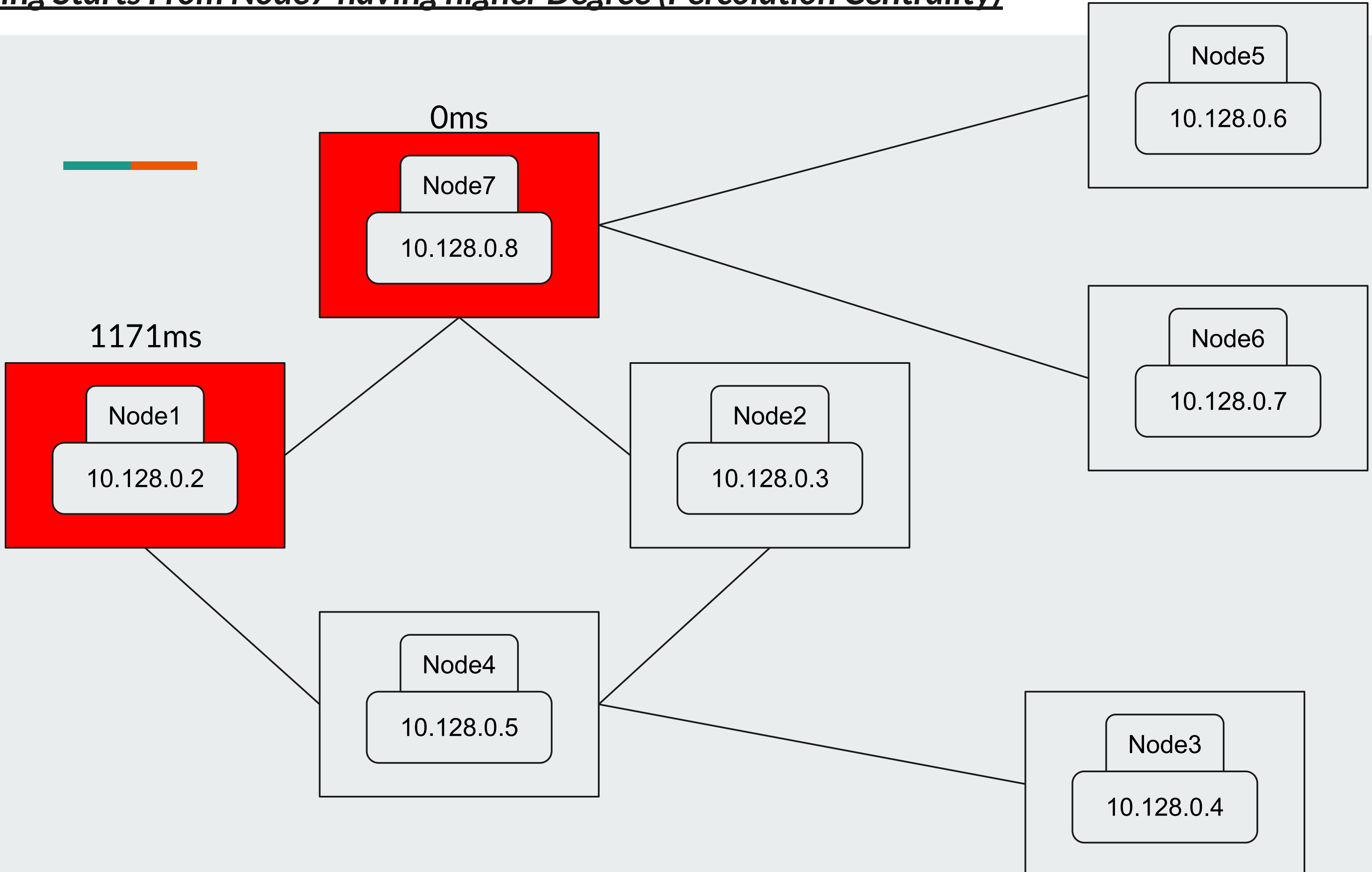
Routing Starts From Node6 having lower Degree (Percolation Centrality)



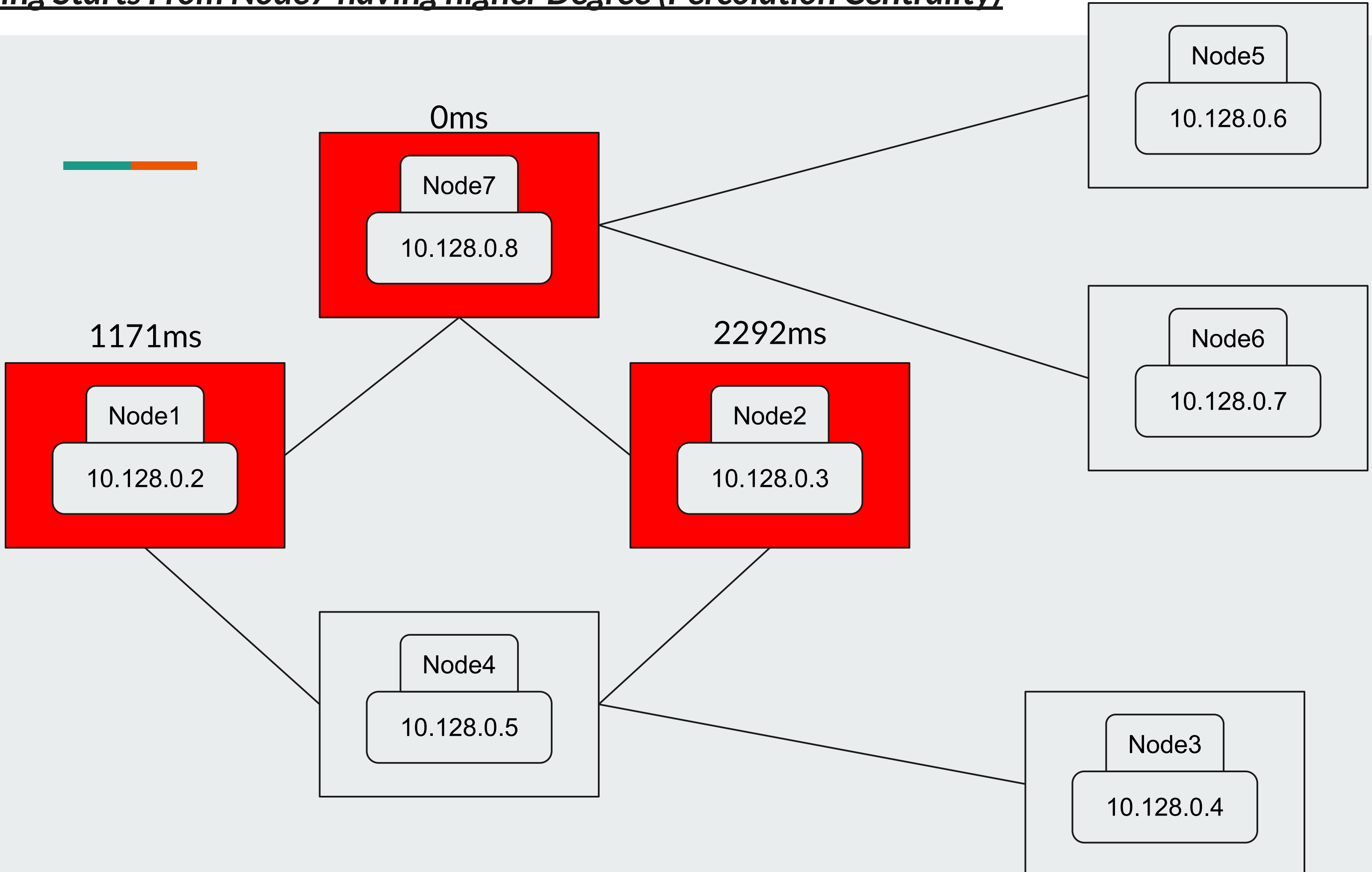
Routing Starts From Node7 having higher Degree (Percolation Centrality)



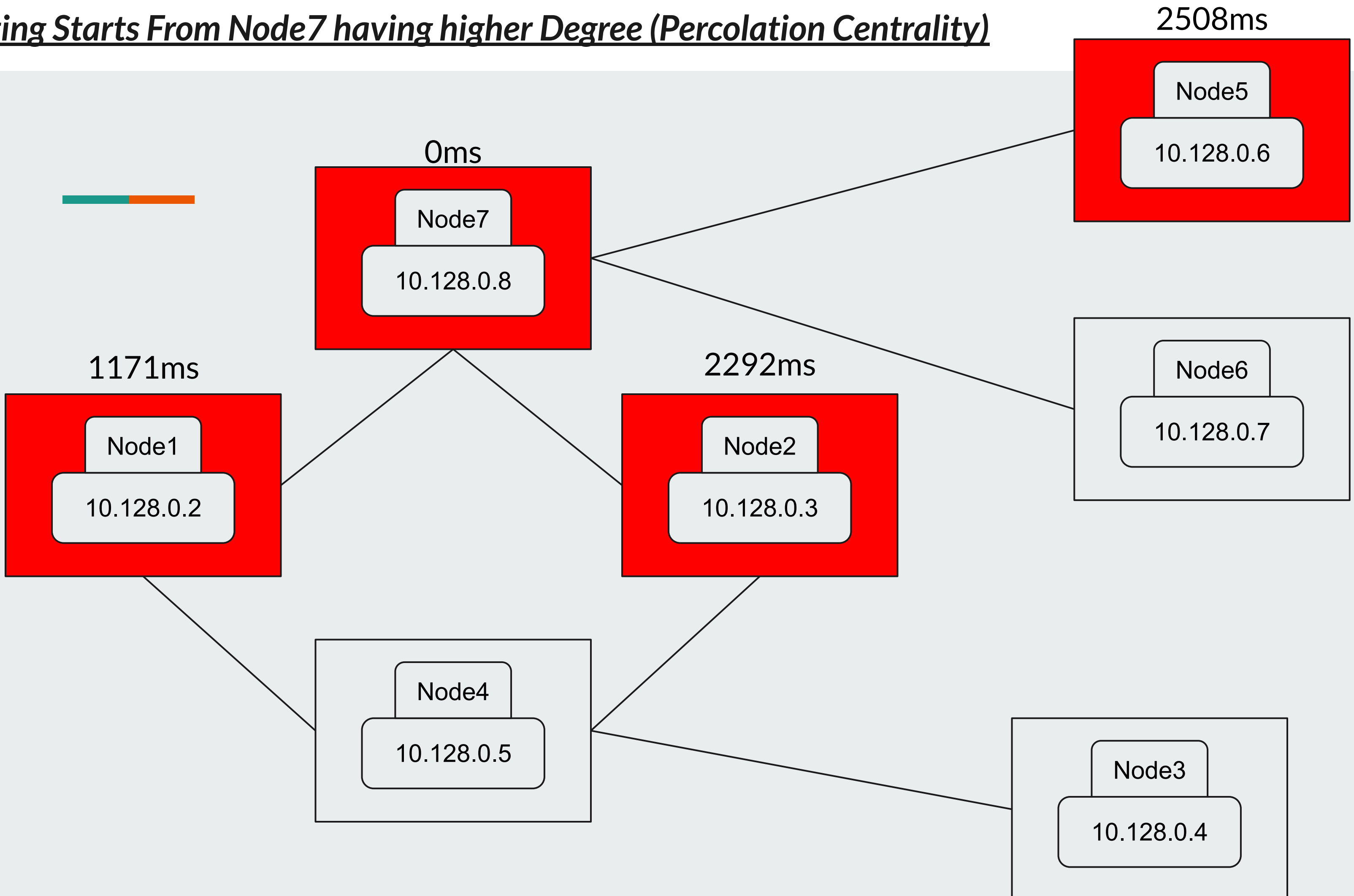
Routing Starts From Node7 having higher Degree (Percolation Centrality)



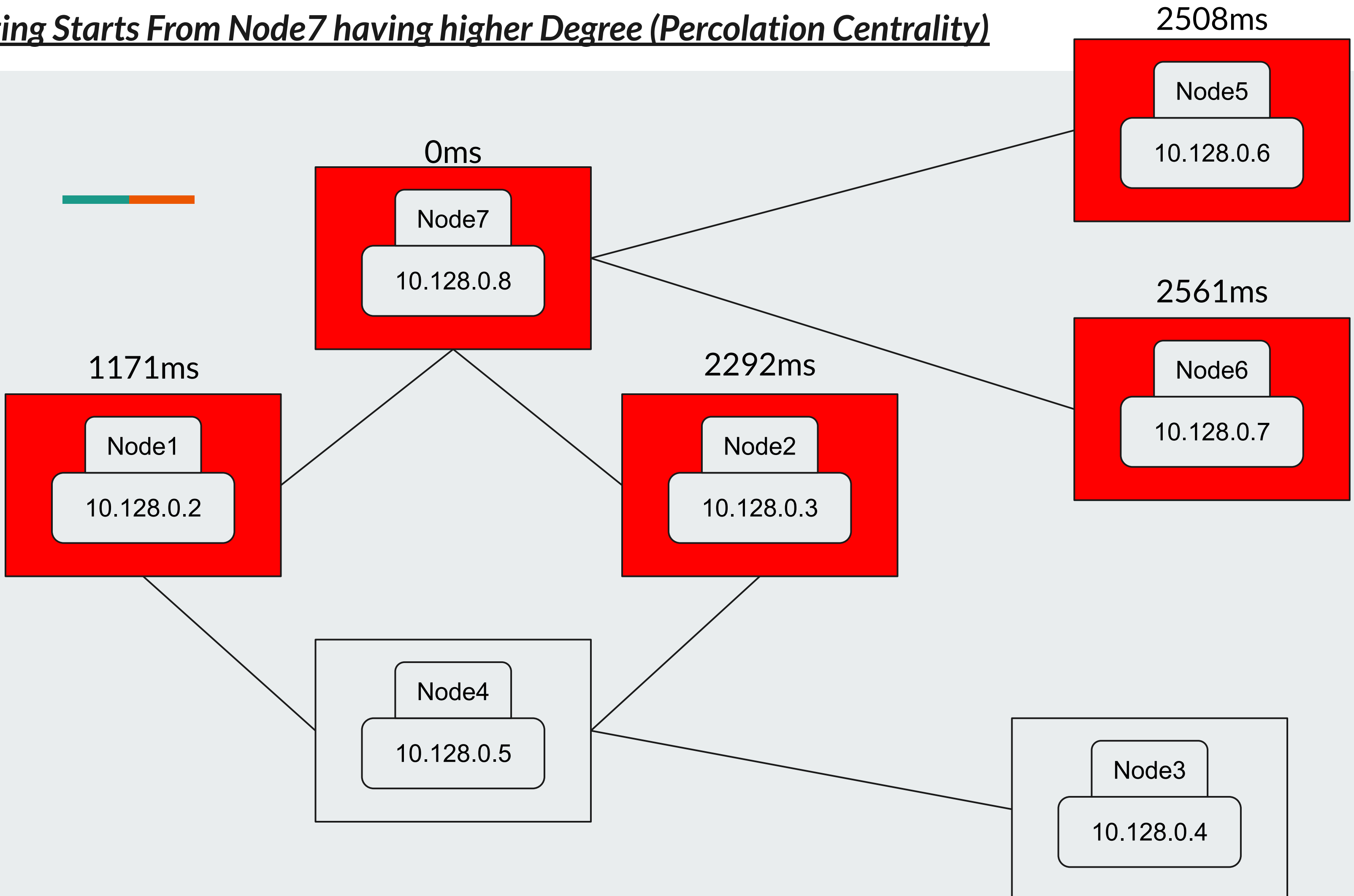
Routing Starts From Node7 having higher Degree (Percolation Centrality)



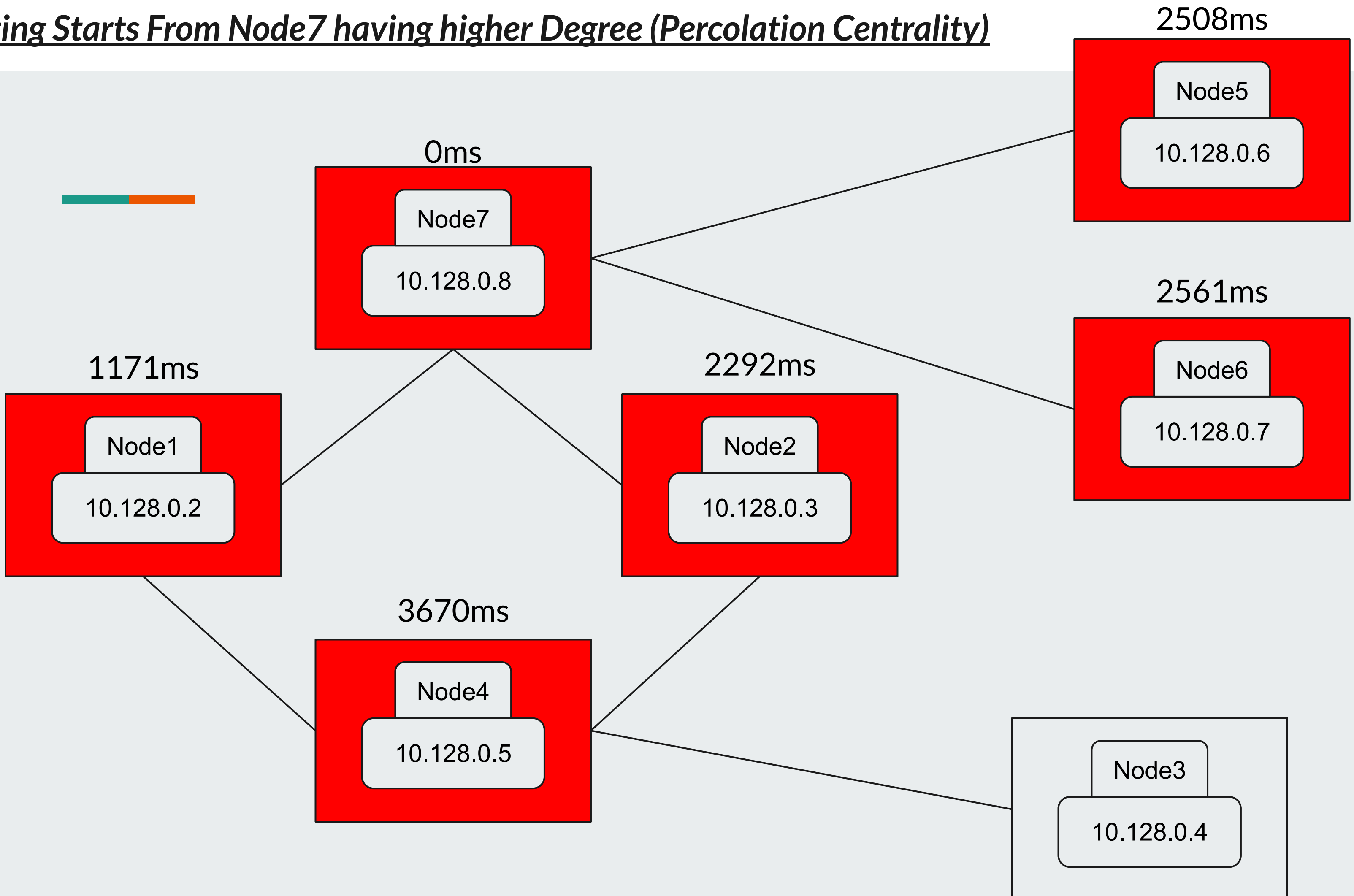
Routing Starts From Node7 having higher Degree (Percolation Centrality)



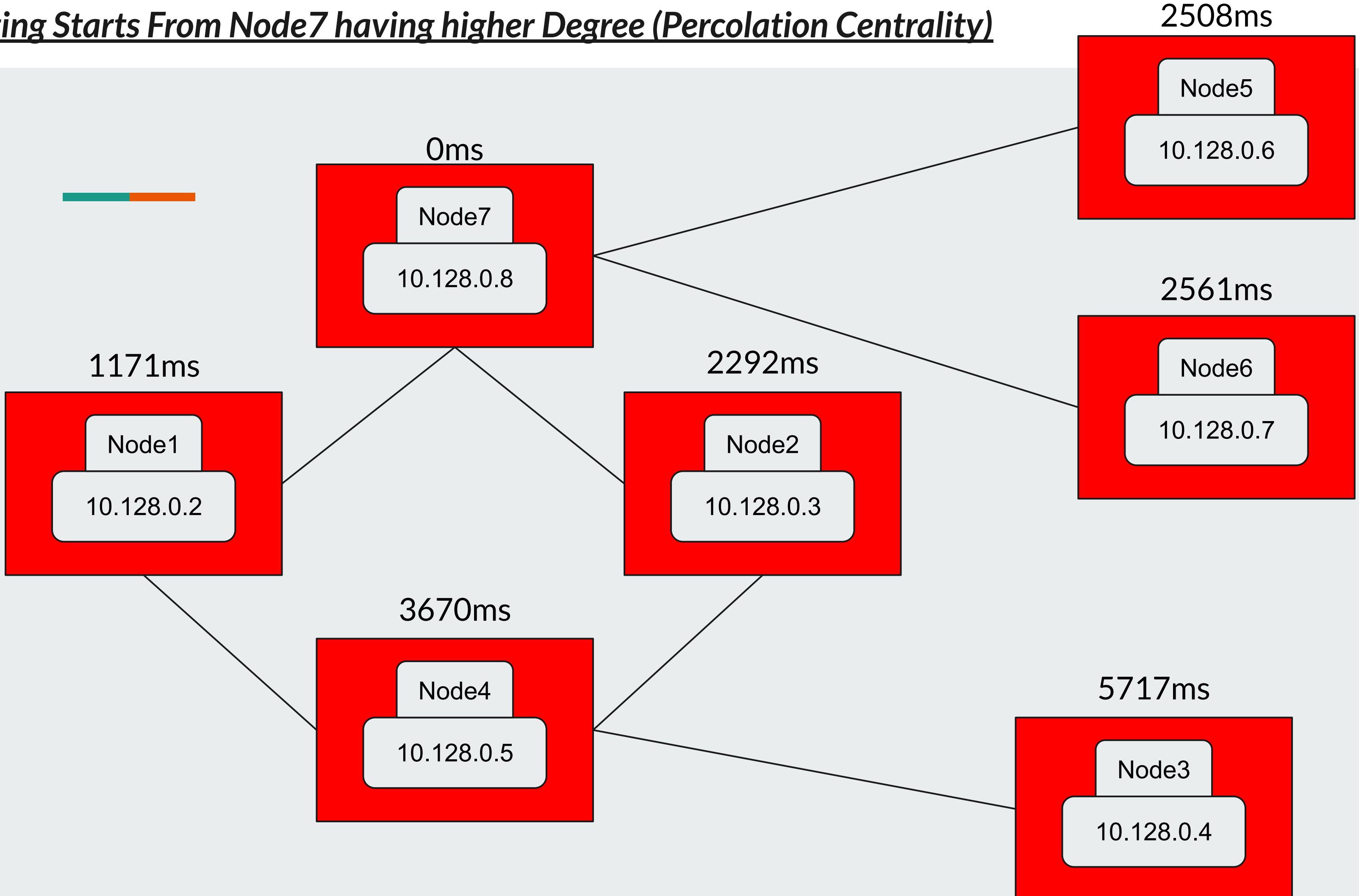
Routing Starts From Node7 having higher Degree (Percolation Centrality)



Routing Starts From Node7 having higher Degree (Percolation Centrality)

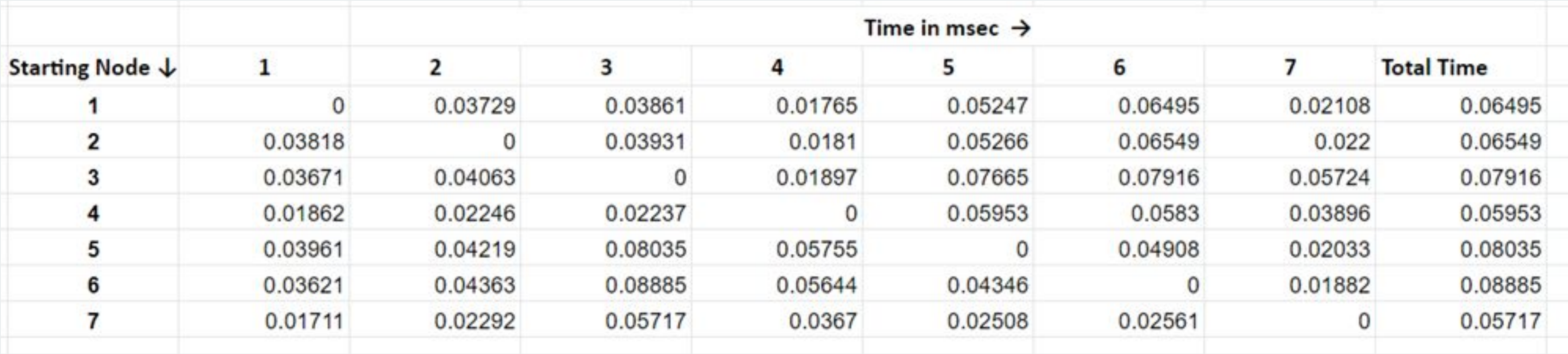


Routing Starts From Node7 having higher Degree (Percolation Centrality)



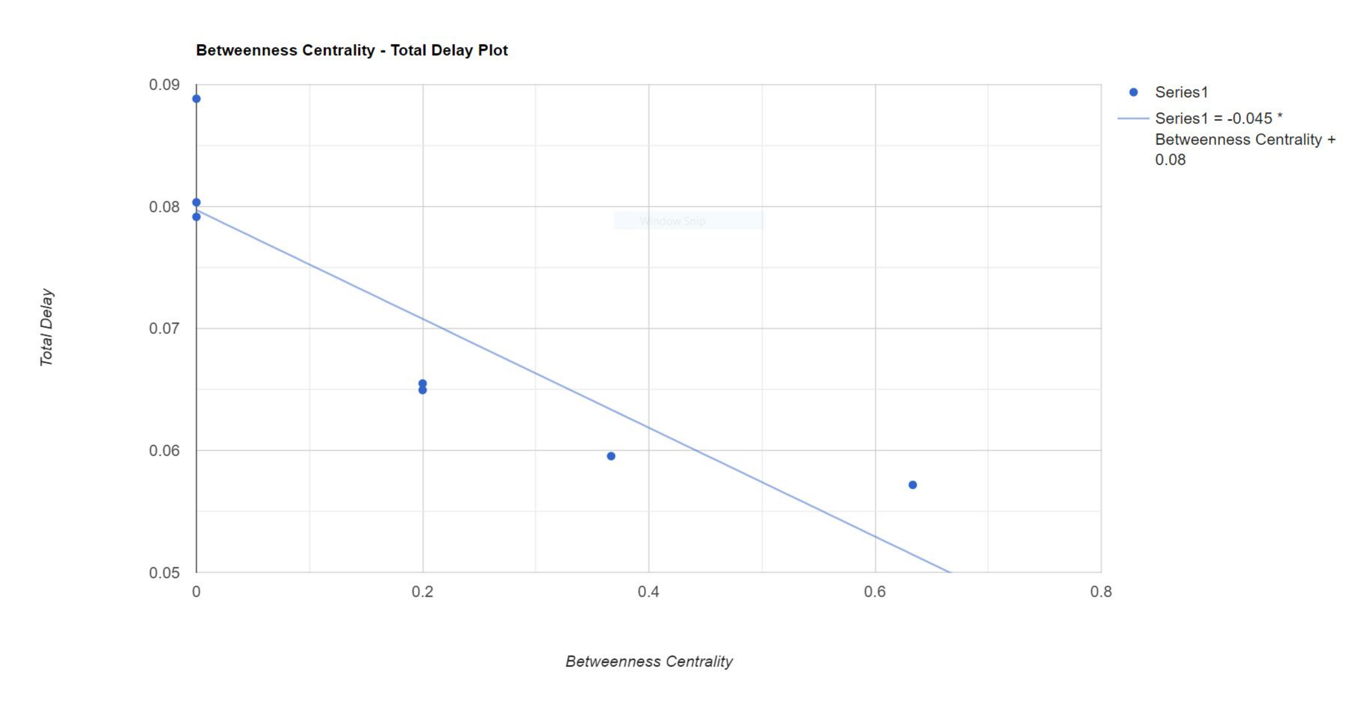
Observation

Time taken for distribution:



					Time in msec →			
Starting Node ↓	1	2	3	4	5	6	7	Total Time
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