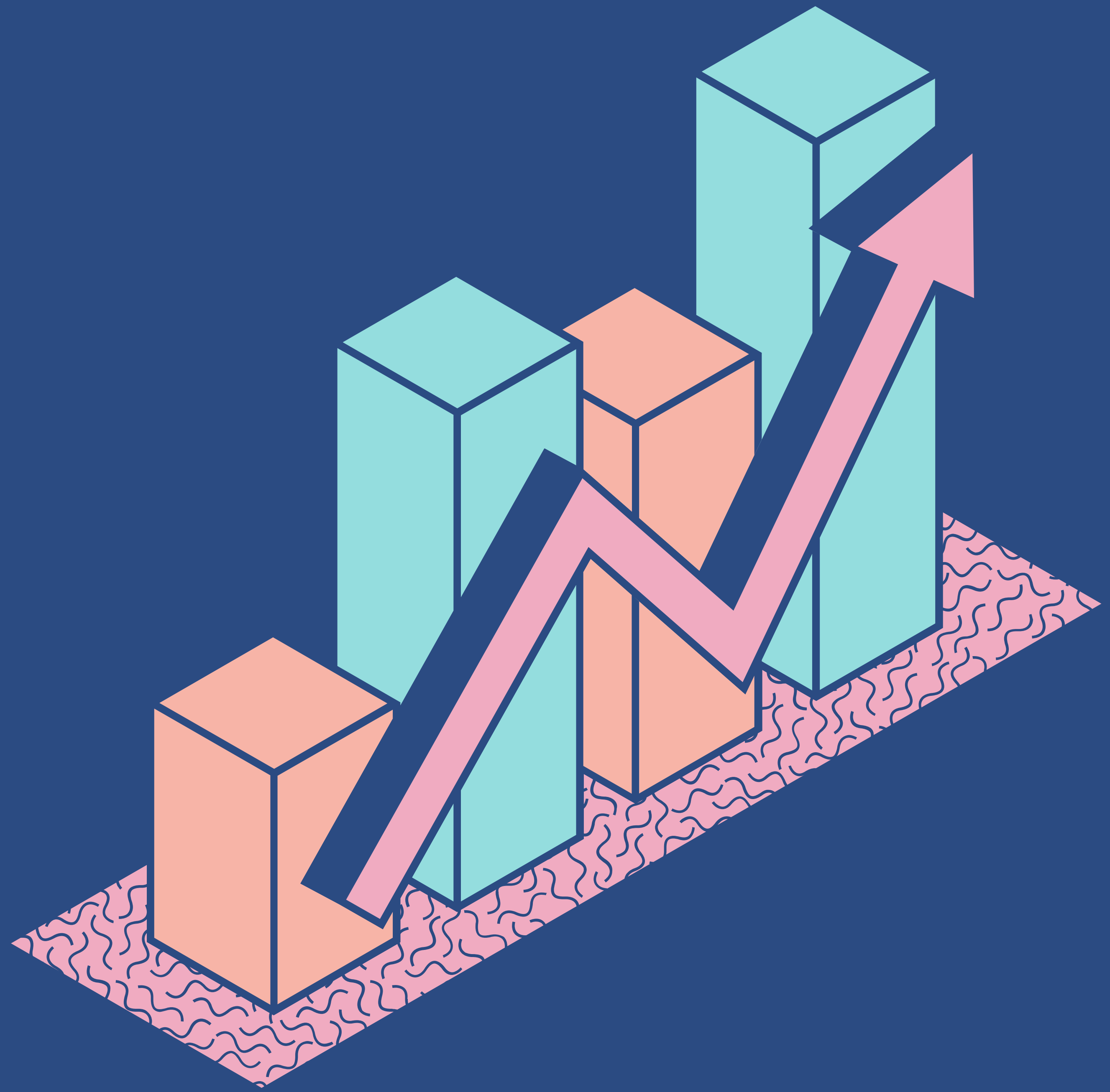
Global Connectivity Framework & 5G Resource Sharing

Abhishek Yadav (1RV21CS009)
Hardik Hiraman Pawar (1RV21CS046)
Harshit Dhoot (1RV21CS049)
Karan Sathish (1RV21CS058)

Index

Understanding the Dataset



Dataset Source

THE DATASET USED IN THIS PROJECT PLAYS A FUNDAMENTAL ROLE IN SHAPING OUR GLOBAL CONNECTIVITY FRAMEWORK AND 5G IOT RESOURCE SHARING ANALYSIS. IT IS DERIVED FROM HTTP://WWW.TOPOLOGY-ZOO.ORG, A REPUTABLE SOURCE FOR NETWORK TOPOLOGIES. THIS DATASET CONTAINS CRUCIAL INFORMATION ABOUT VARIOUS NETWORK NODES, ENABLING US TO DEFINE AND ANALYZE NETWORK STRUCTURES IN DIFFERENT REGIONS.

Regions Considered

BT Asia Pacific Region

- This region encompasses nodes from diverse Asian and Oceanic countries, including Indonesia, Australia, Singapore, Malaysia, and more.
- It provides detailed information about each node's country, longitude, and latitude, forming the foundation for our analysis.

Quest Region

- The Quest region defines nodes based on data obtained from an external source. It includes nodes within the Quest network.
- Information such as country, longitude, and latitude is used to establish the nodes within this network.

Regions Considered

TATA Region

- Similarly, the TATA region defines nodes based on data from an external source, representing the TATA network.
- The dataset specifies crucial details, including the country, longitude, and latitude of nodes within this network.

ERNET Region

- Focused on India, the ERNET region incorporates nodes from cities like Pune, Indore, Trivandrum, Mumbai, and more.
- It provides comprehensive insights into each node's country, longitude, and latitude, aiding our analysis of this region.

Regions Considered

PERN Region

- The PERN region features nodes in Pakistan, including cities like Peshawar, Karachi, Faisalabad, Quetta, Lahore, and others.
- Similar to other regions, it furnishes vital data about each node's country, longitude, and latitude.

Literature Survey

[1] Research on the Construction of Resource Sharing Platform Based on MicroService - Wang Xiaojun, Sun Xu; Hao Zhe, Li Huijuan

This paper discusses the integration of Internet technology and education, emphasizing the importance of a resource sharing platform for lifelong education in Jiangsu. It proposes a framework for such a platform using technologies like cloud computing and big data, outlines its functions at seven levels, offers technical solutions, and addresses potential issues with suggestions for improvement.

Literature Survey

[2] Resource sharing in 5G mmWave cellular networks - Mattia Rebato, Marco Mezzavilla, Sundeep Rangan, Michele Zorzi

This paper discusses the importance of resource sharing in mmWave network design to fully utilize bandwidth and accommodate variable traffic. It explores different sharing configurations, considering technical and economic aspects. The study highlights the impact of mmWave channel characteristics and antenna properties on results and suggests that full spectrum and infrastructure sharing can benefit both service providers economically and improve user rates.

Literature Survey

[3] Generalized resource sharing - Raje, Bergamaschi

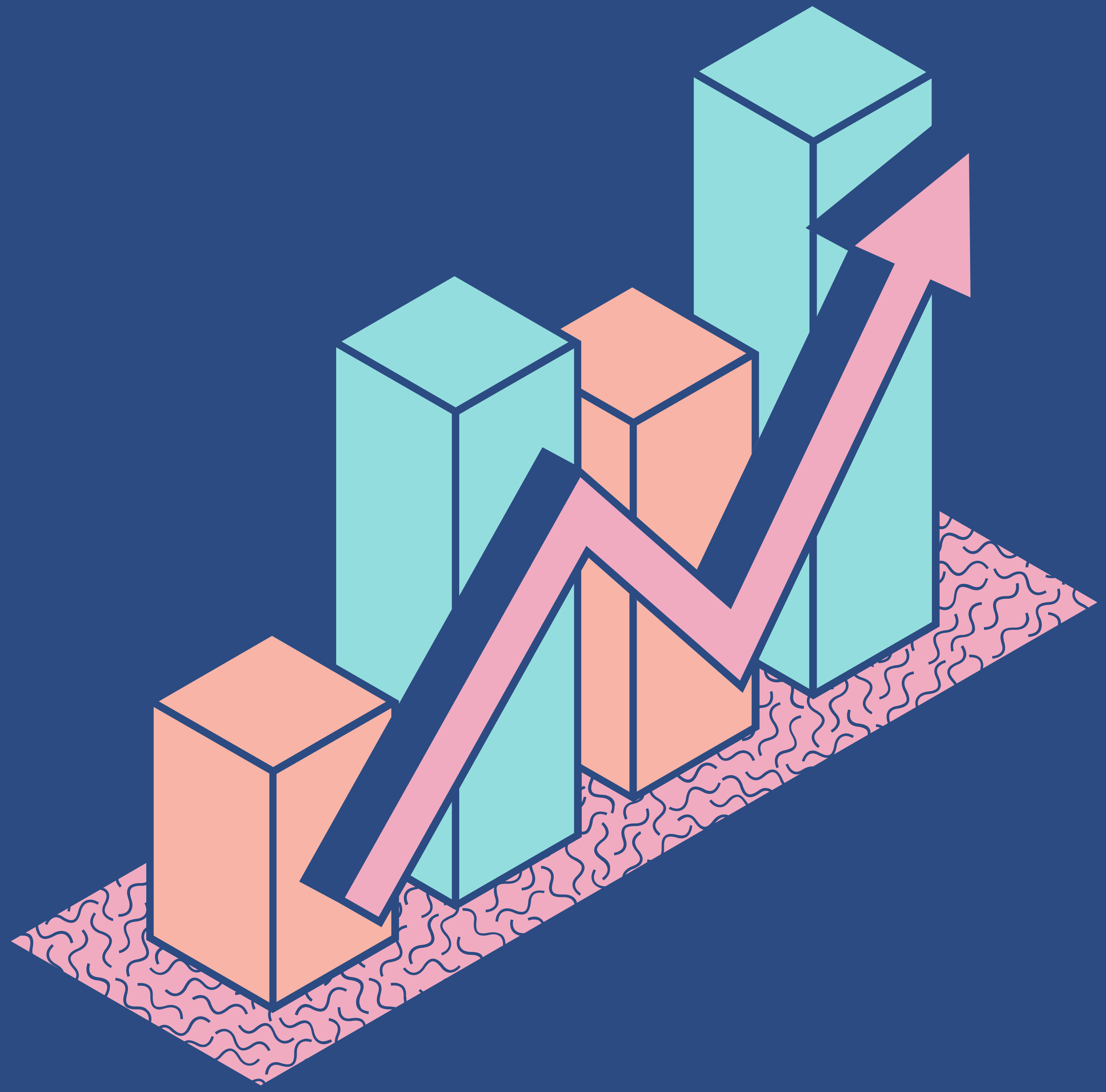
This paper introduces improved algorithms for resource sharing in high-level synthesis, addressing limitations in existing methods. These algorithms use a global clique partitioning approach with accurate cost estimation, considering registers and functional units. The results demonstrate reduced design delay and minimized area, particularly for larger designs with multiple sharing possibilities.

Literature Survey

[4] Flow Control in a Resource-Sharing Computer Network - R. Kahn, W. Crowther

This paper explores flow control in a resource-sharing computer network with geographically distributed, interconnected computers. It focuses on how messages flow efficiently through the communication subnet and its relationship with host flow control and overall subnet performance.

Calculating and Visualizing Latency Data



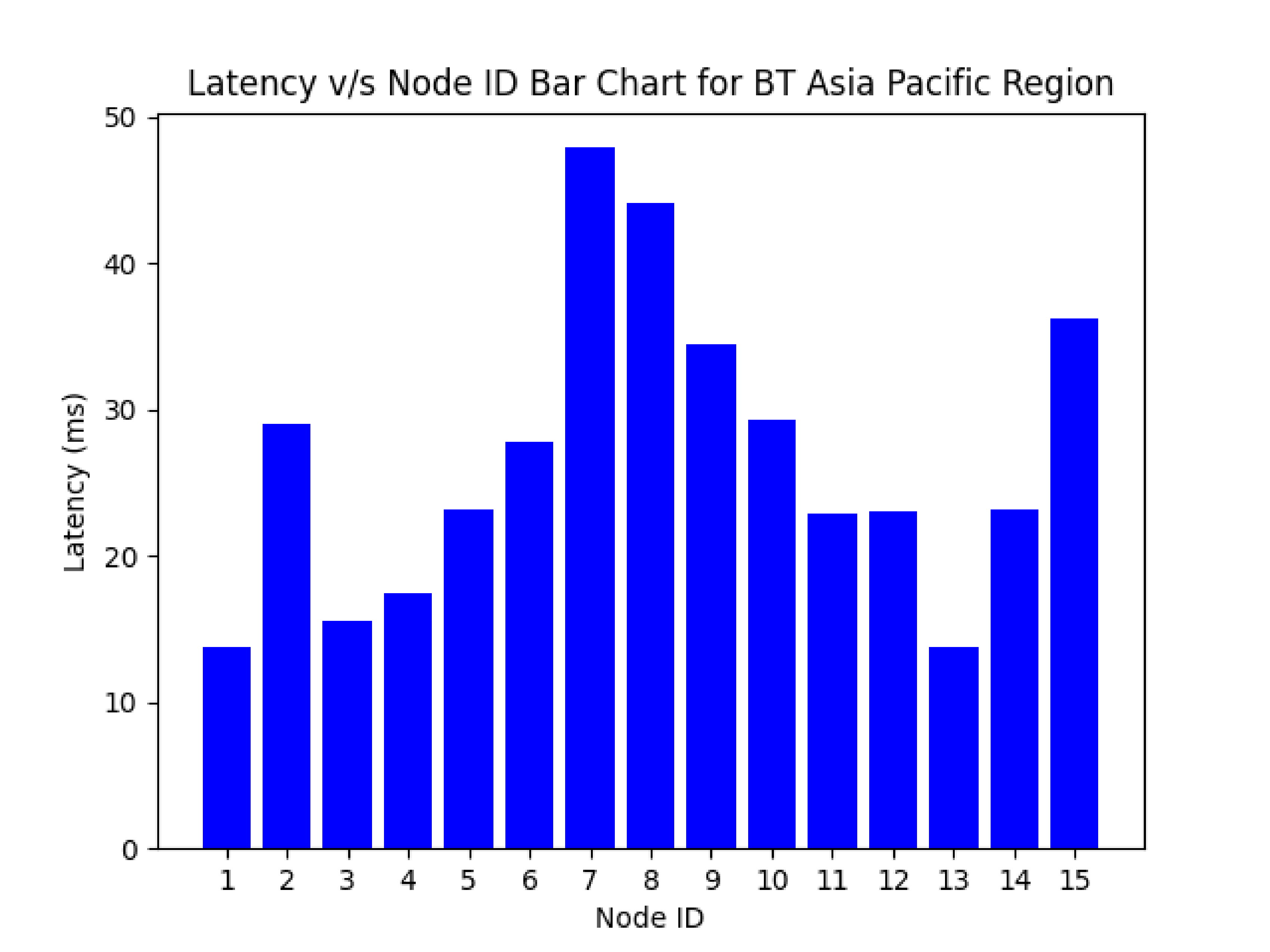
Latency Calculation

- We compute the latency (delay) for each network node using the formula: Latency = 2 * distance / speed.
- Latitude and longitude information is leveraged to determine the distance between nodes, while the speed of data transmission is set at 300,000 km/s.

```python
def getLatency(long, lat, mid_long, mid_lat):

    long_km = long * 111.32 * math.cos(math.radians(lat))
    lat_km = lat * 110.574

    mid_long_km = mid_long * 111.32 * \
        math.cos(math.radians(mid_lat))
    mid_lat_km = mid_lat * 110.574

    distance = math.sqrt((long_km - mid_long_km)**2 +
                         (lat_km - mid_lat_km)**2)     # in km

    speed = 3e5         # in km/s

    time = (2 * distance / speed) * 1000  # in ms

    return time
```

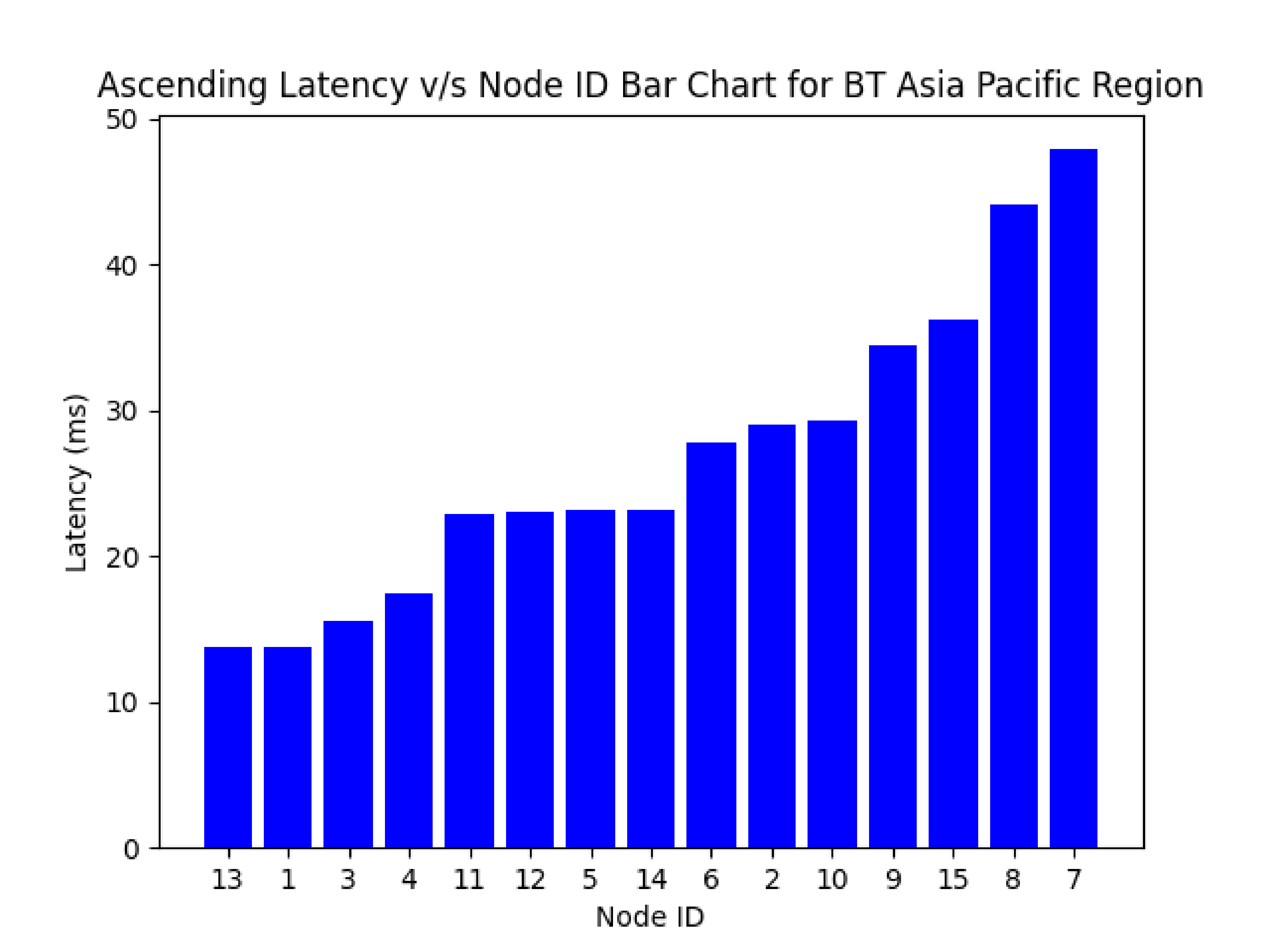
Graph Visualization

- We visualize latency data through a series of informative graphs.
- Initially, we plot the latency graph in its natural order, offering insights into the latency distribution among network nodes.



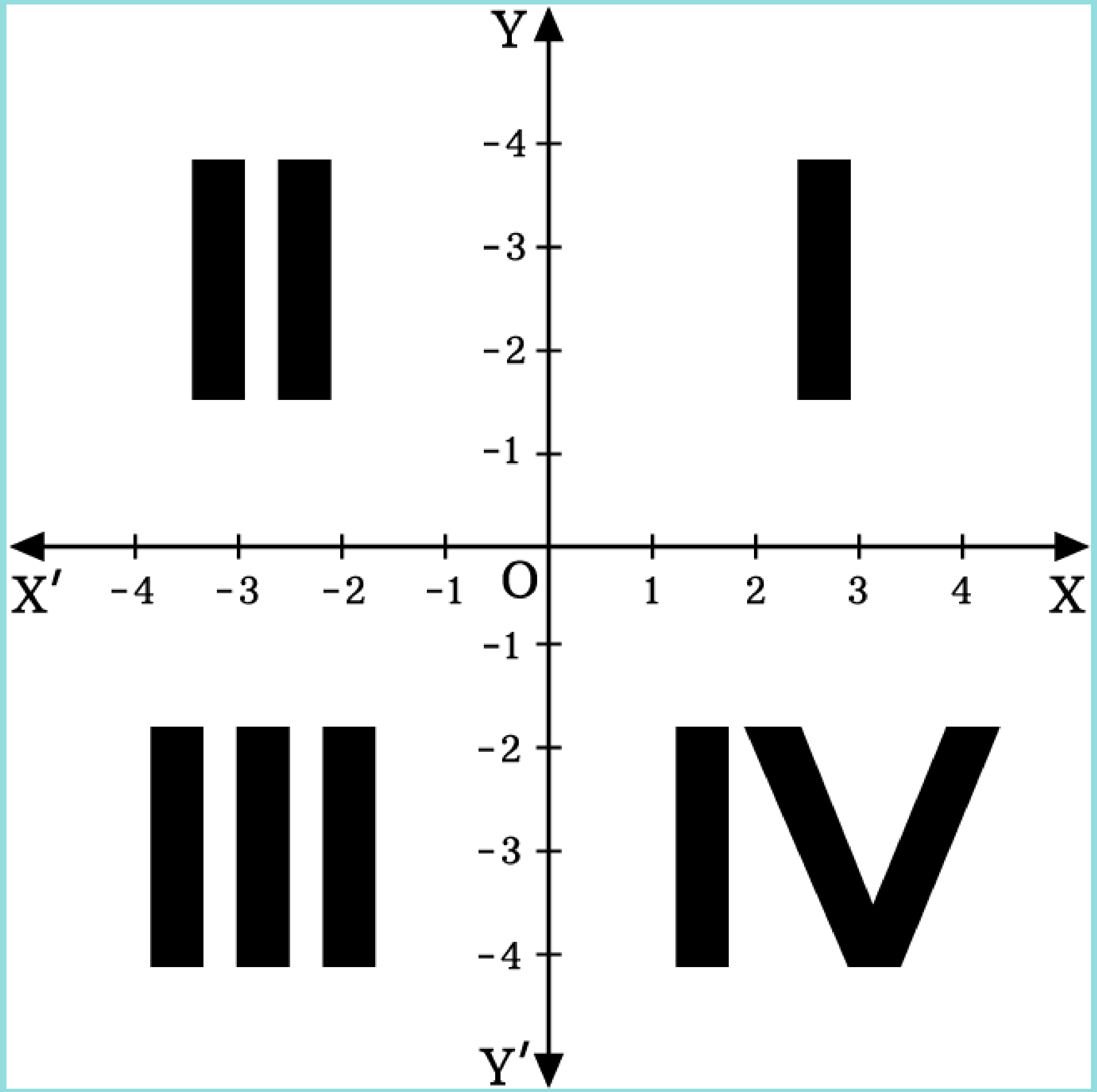
Ascending Latency Order

- To facilitate a more detailed analysis, we rearrange the latency data in ascending order.
- This enables us to identify nodes with the lowest and highest latencies, aiding in load distribution decisions.

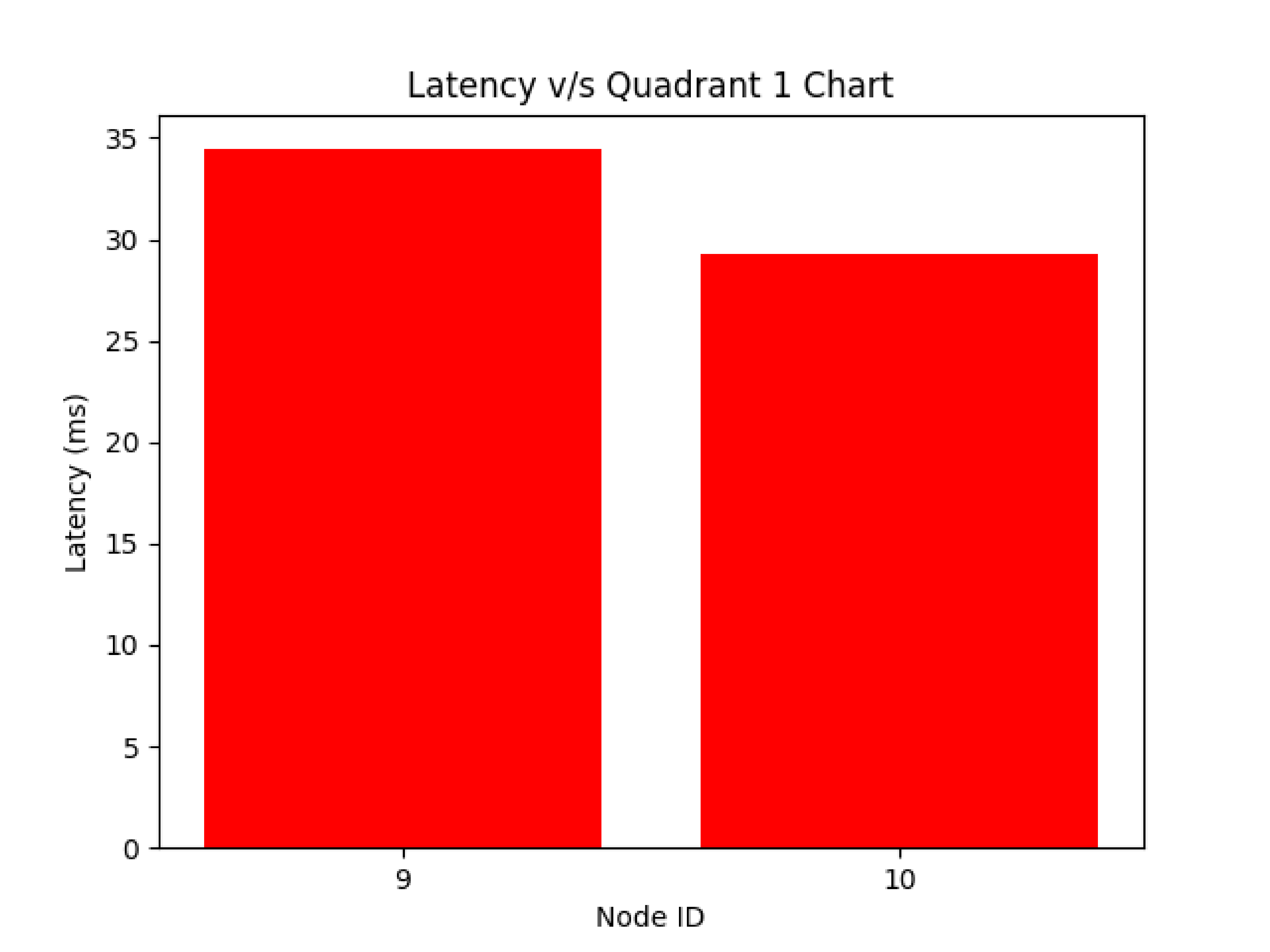


Clustering Nodes

- The network nodes are grouped into four quadrants, assuming the midpoint as the origin.
- This clustering approach allows us to understand the spatial distribution of nodes and their respective latencies within each quadrant.

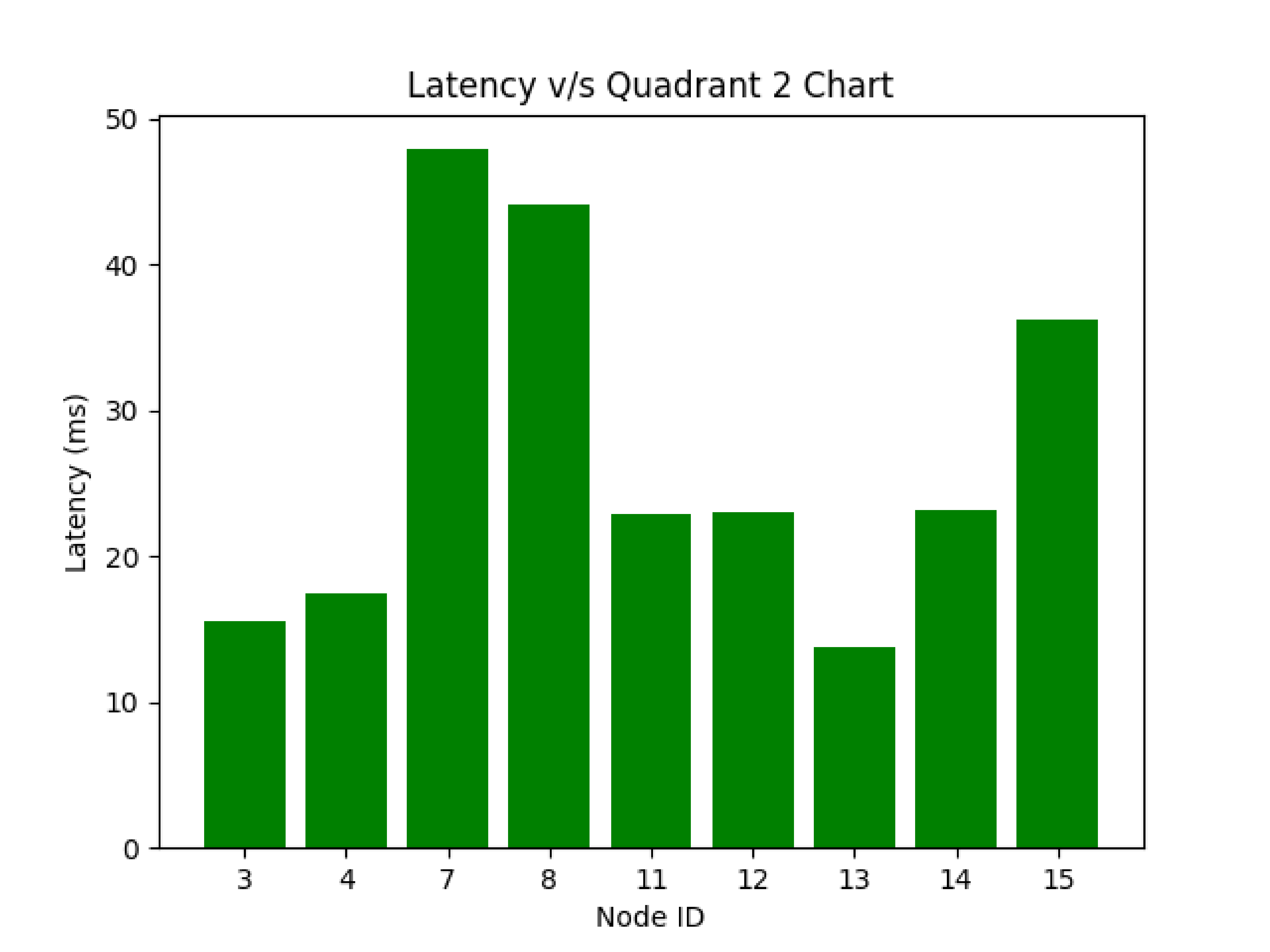


Quadrant 1

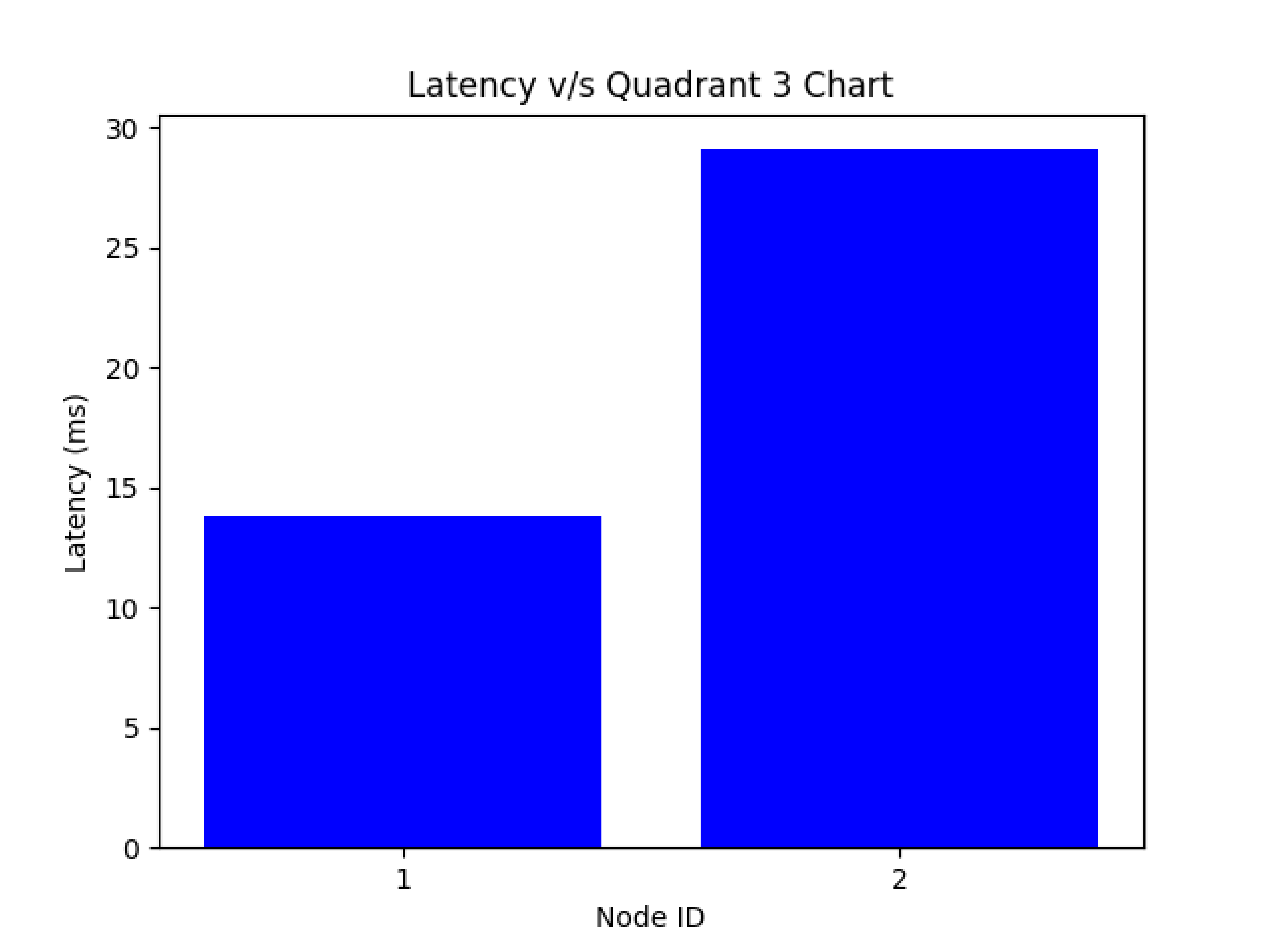


Latency v/s Quadrant 1 Chart
Latency (ms)
35
30
25
20
15
10
5
0
9
10
Node ID

Quadrant 2

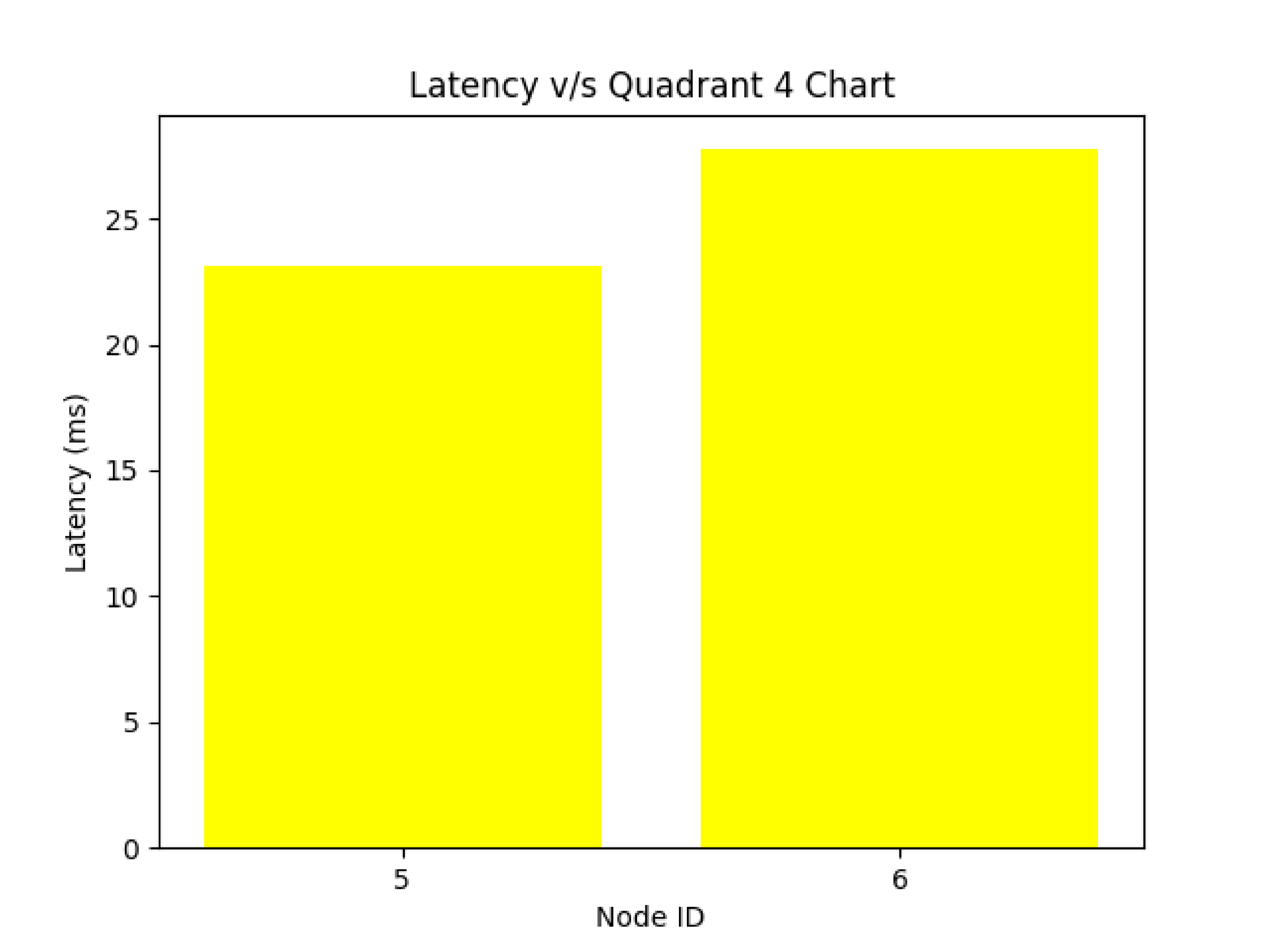

Latency v/s Quadrant 2 Chart
Latency (ms)
0
10
20
30
40
50
3
4
7
8
11
12
13
14
15
Node ID

Quadrant 3



Latency v/s Quadrant 3 Chart
Latency (ms)
30
25
20
15
10
5
0
1
2
Node ID

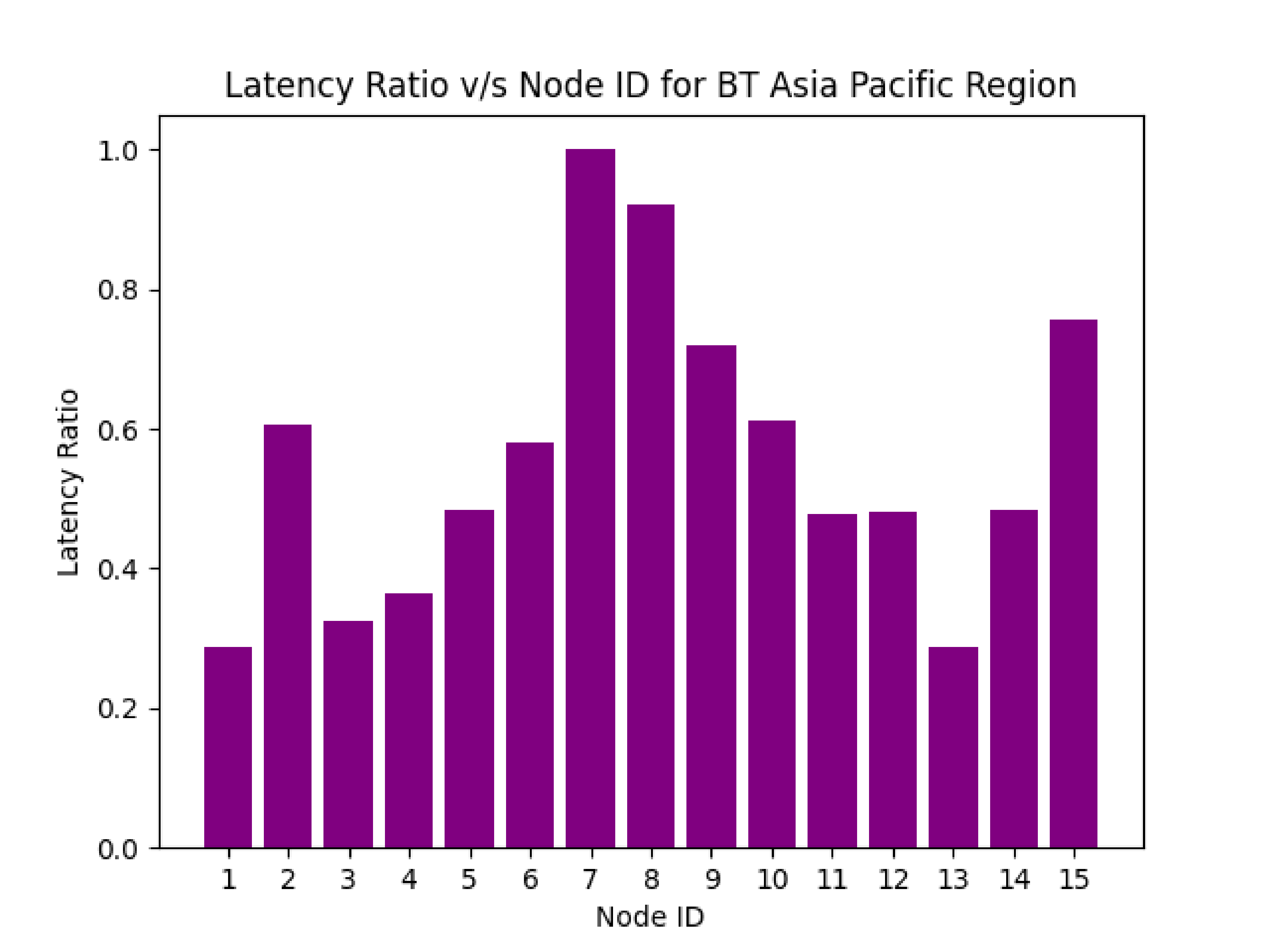
Quadrant 4



Latency v/s Quadrant 4 Chart
Latency (ms)
0
5
10
15
20
25
5
6
Node ID

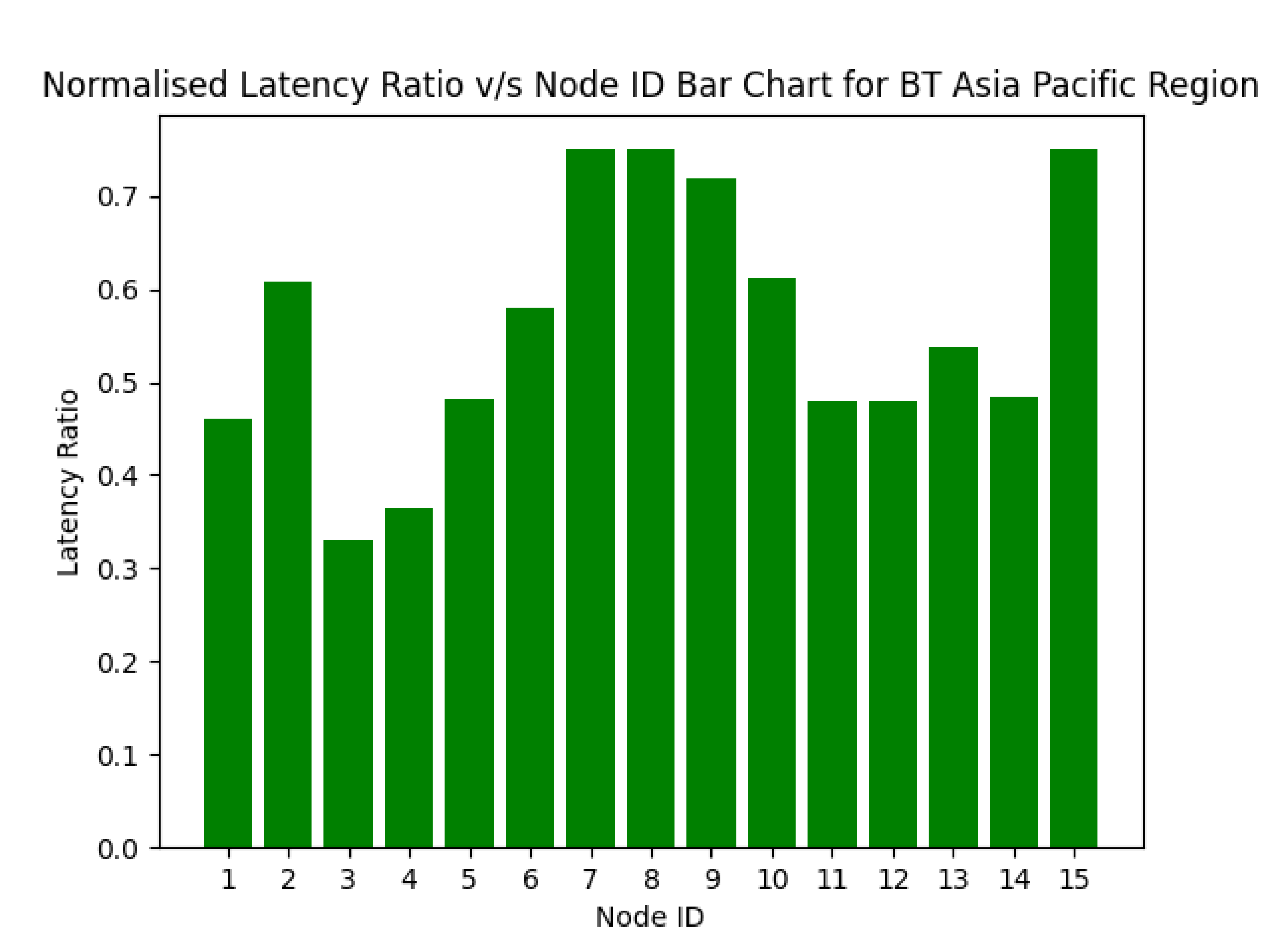
Latency Ratio Analysis

- We normalize latency values to a range between 0 and 1 for further analysis.
- The normalized latency ratios help identify nodes with the greatest impact on latency and potential resource sharing improvements.

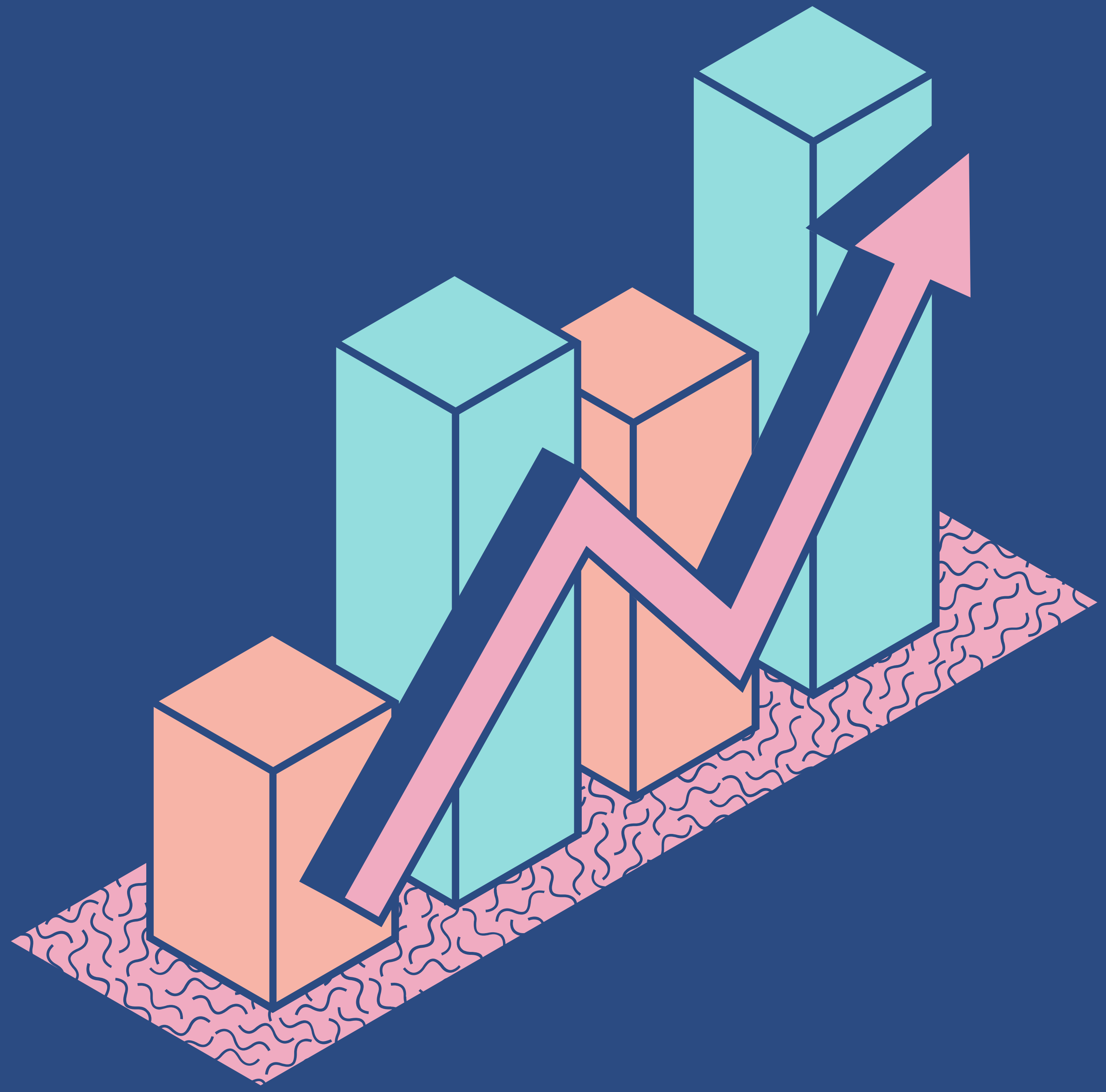


Load Balanced Latency Ratio

- The nodes with latency greater than or equal to a certain threshold are identified and the load is balanced to the nodes with the least load.
- This in turn, helps lessen congestion and improve throughput of the network system.



Calculating and Visualizing Load DATA



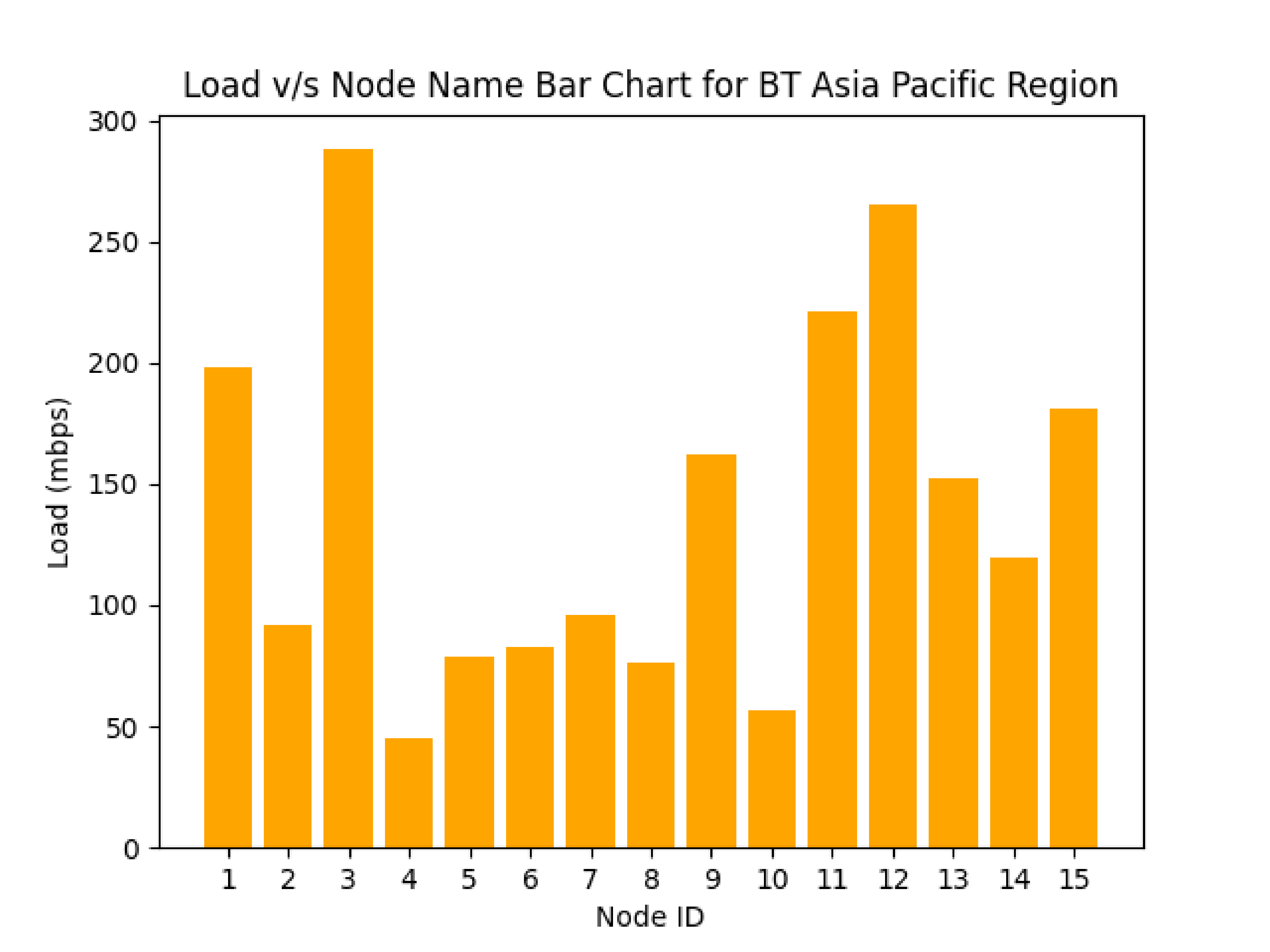
Load Generation

To simulate real-world network conditions, we randomly generate load values for each node within specified lower and upper limits. These load values represent the amount of data traffic that each node can handle, measured in megabits per second (Mbps).

```python
def getLoad(lower, upper):
    """Returns the load between two points in mbps.

    Args:
        lower (int): Lower limit of the load. (mbps)
        upper (int): Upper limit of the load. (mbps)

    Returns:
        load (int): Load between the two points in mbps.
    """

    return random.randint(lower, upper)
```

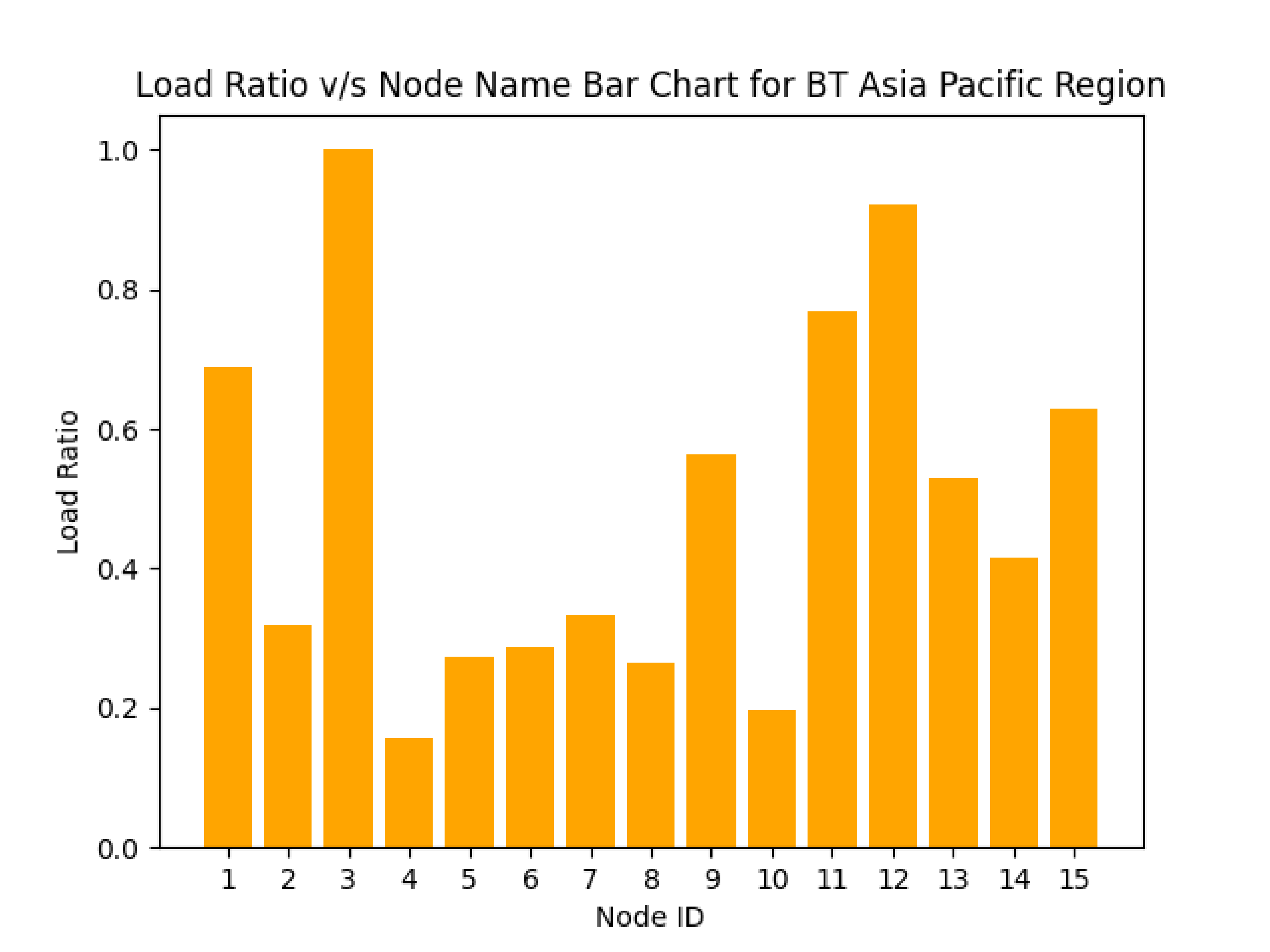
Load Ratio Visualization

We plot a bar chart that visualizes the load values for each node. This provides an overview of the distribution of load across the network, helping us identify nodes that may be underutilized or overloaded.



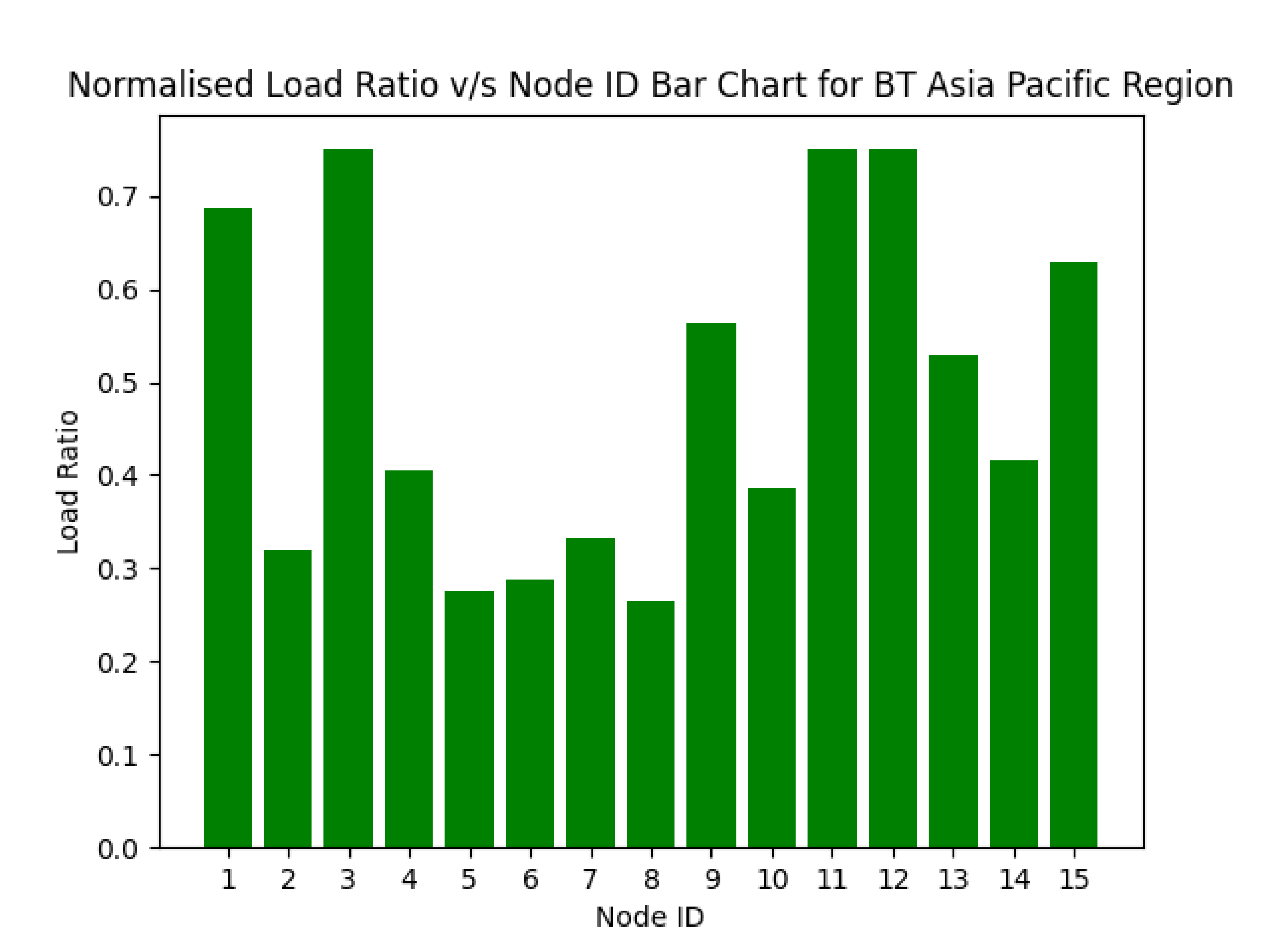
Load Ratio Normalization

To facilitate meaningful comparisons, we normalize the load values to a range between 0 and 1. This normalization ensures that even if the load values vary widely, they can be compared on a consistent scale.

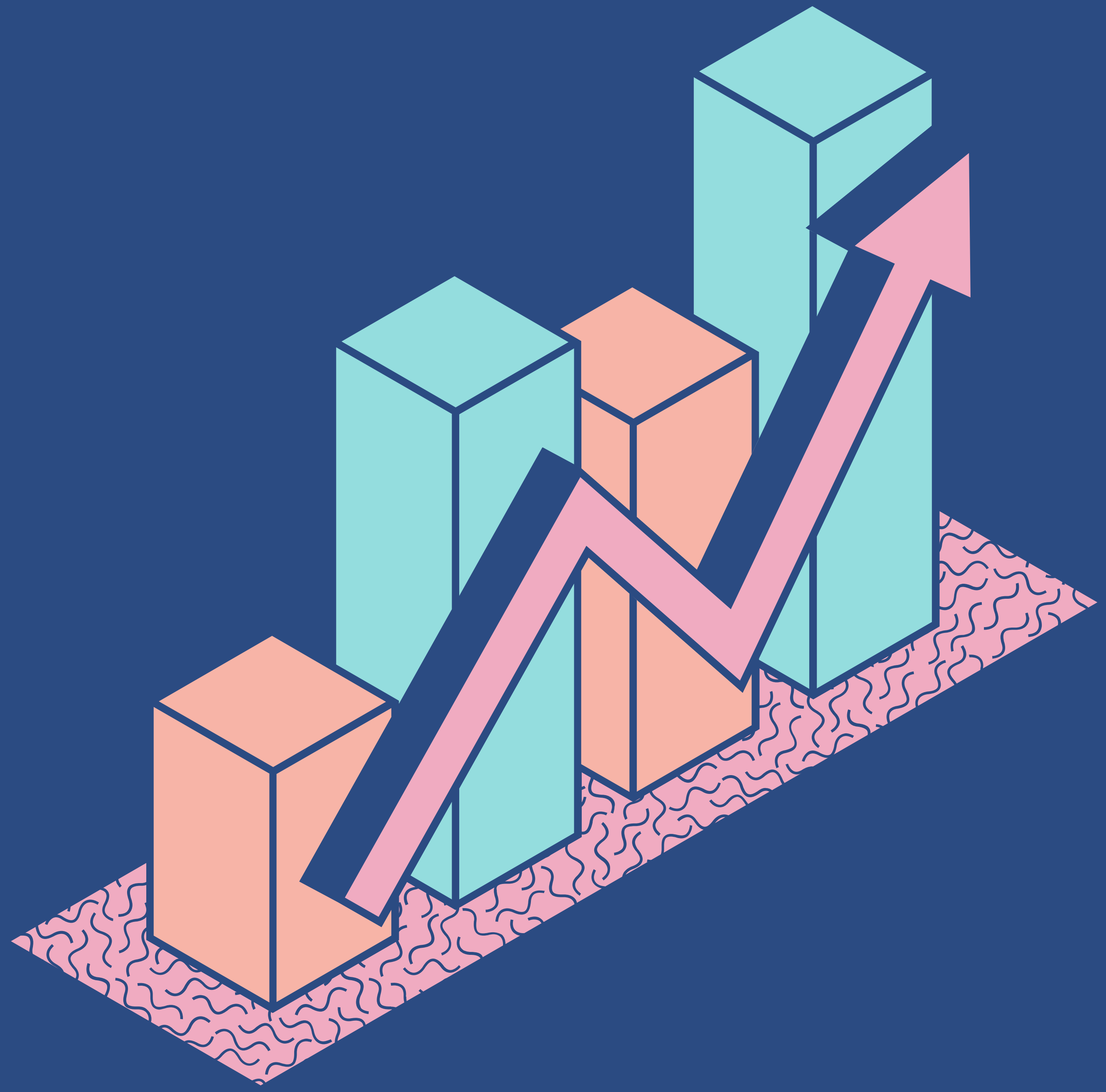


Load Balancing

Load balancing is a crucial aspect of network optimization. We aim to balance the load across nodes to prevent congestion and ensure efficient data transmission. Nodes with load ratios exceeding a defined threshold are considered for load redistribution.



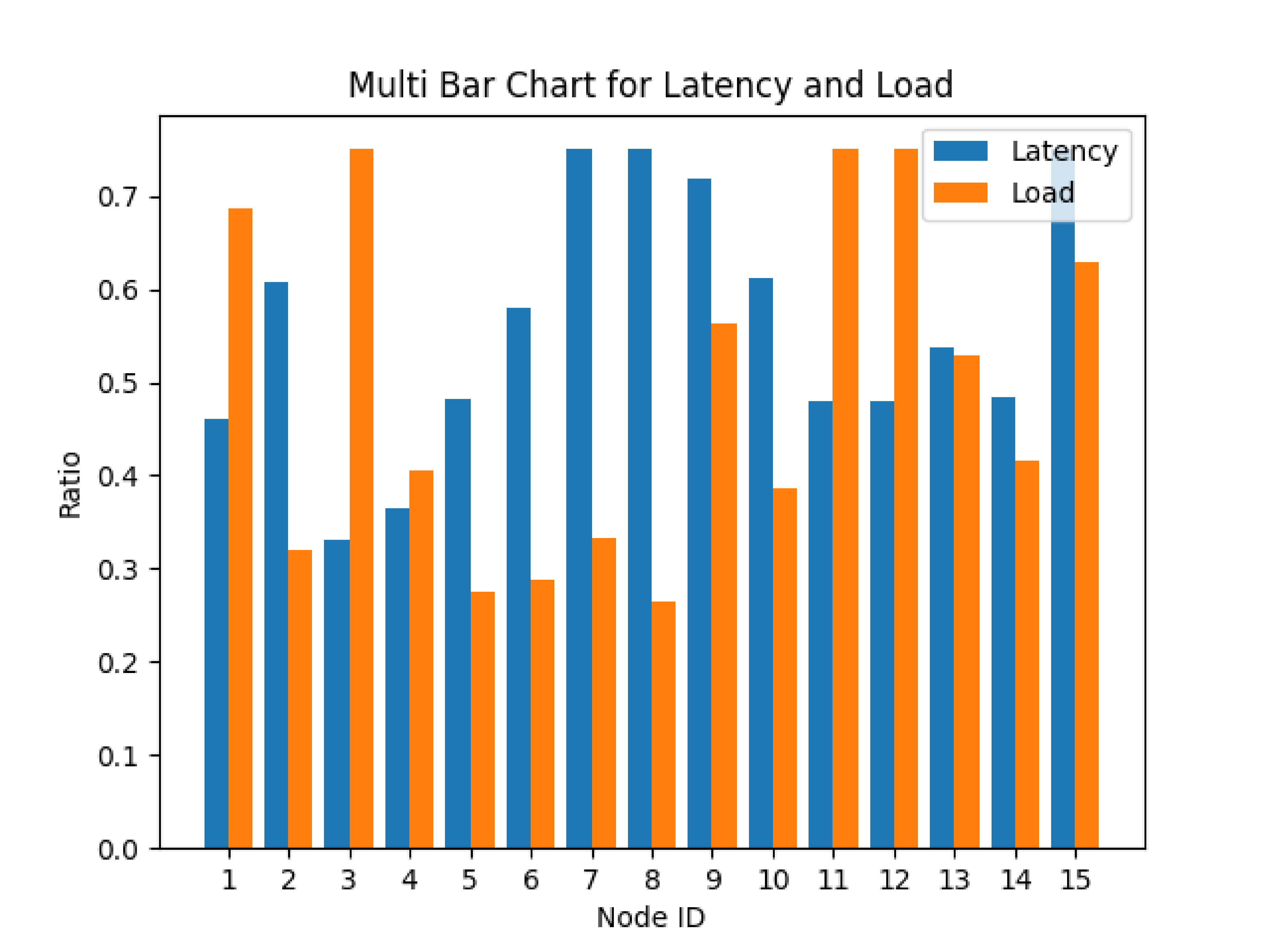
Multi Bar Chart for Combined Analysis of Latency and Load



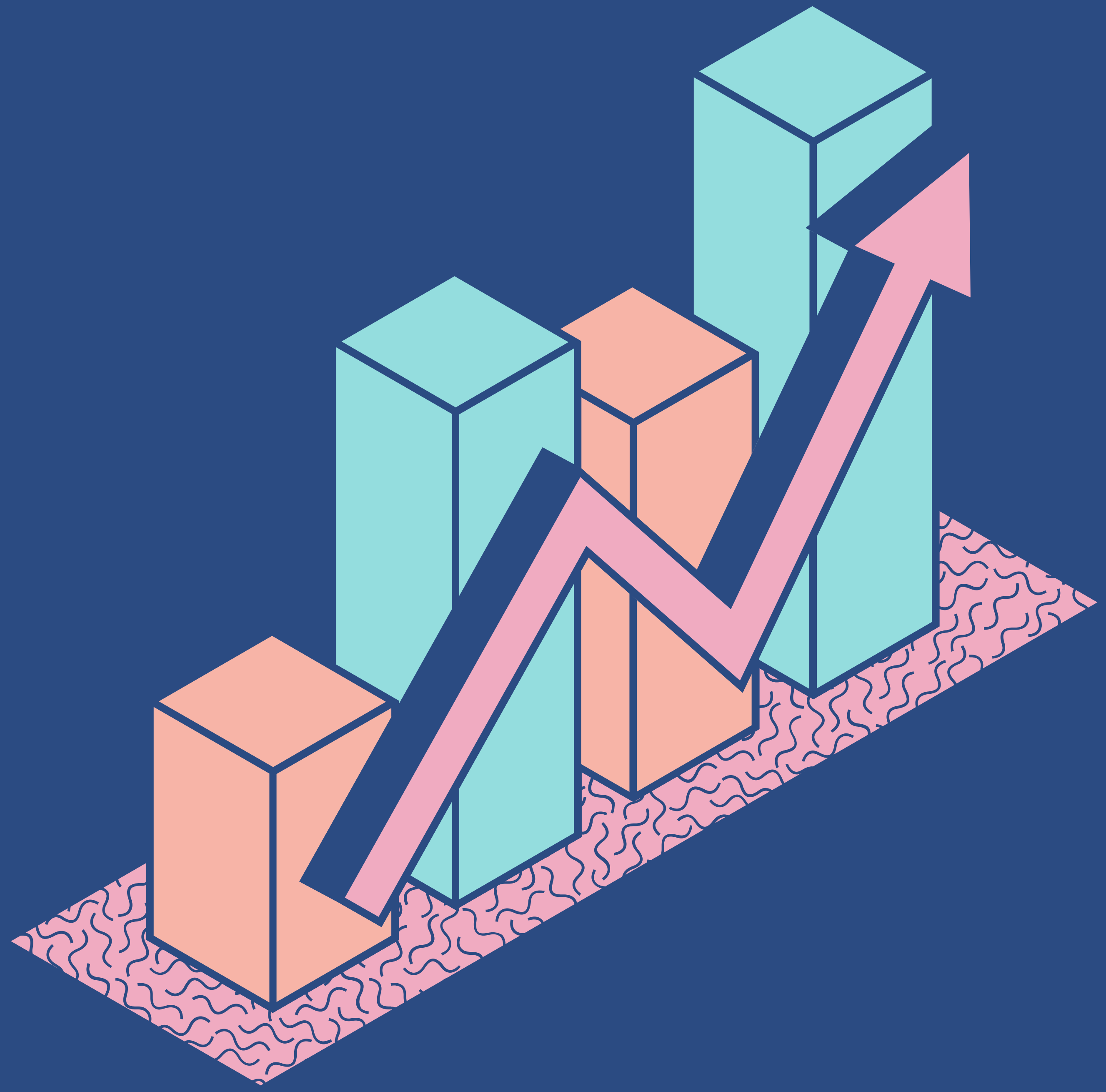
Combined Analysis

- The multi-bar chart combines latency and load ratio data for each node. Latency is represented by one set of bars, while load ratios are represented by another set. This combination allows for a comprehensive assessment of network performance.
- The chart is structured with nodes on the X-axis and latency and load ratio on the Y-axis. Latency and load ratios are visually represented for each node, making it easy to identify nodes with latency or load issues.
- This multi-bar chart provides a holistic view of network performance, highlighting nodes that may require optimization or load balancing. It serves as a valuable tool for network administrators and engineers to make informed decisions about network management and improvements.

Multi Bar Chart Latency & Load Visualization



Applications and Conclusion



Applications

Network Optimization

The project can be applied to optimize computer networks, data centers, or cloud infrastructures. By analyzing latency and load data, network administrators can identify bottlenecks and make necessary adjustments to improve overall network performance.

Content Delivery Networks (CDNs)

CDNs can use this project to strategically place content servers based on network latency and load. This ensures faster content delivery to end-users and a more efficient use of resources.

Applications

Traffic Management

Traffic management systems can benefit from this project by optimizing routing decisions. By considering both latency and load, traffic can be directed through paths that minimize delays and prevent network congestion.

Gaming Networks

Online gaming networks require low-latency connections. This project can help gaming platforms allocate resources to players in a way that minimizes latency, enhancing the gaming experience.

Conclusion

IN CONCLUSION, THIS PROJECT OFFERS A COMPREHENSIVE SOLUTION FOR ANALYZING AND OPTIMIZING NETWORK PERFORMANCE BY CONSIDERING BOTH LATENCY AND LOAD. BY VISUALIZING THESE KEY METRICS IN A MULTI-BAR CHART, NETWORK ADMINISTRATORS GAIN VALUABLE INSIGHTS INTO THE HEALTH OF THEIR NETWORKS AND CAN MAKE DATA-DRIVEN DECISIONS FOR IMPROVEMENTS.

References

Research Papers

[1] Research on the Construction of Resource Sharing Platform Based on MicroService - Wang Xiaojun, Sun Xu; Hao Zhe, Li Huijuan

[2] Resource sharing in 5G mmWave cellular networks - Mattia Rebato, Marco Mezzavilla, Sundeep Rangan, Michele Zorzi

[3] Generalized resource sharing - Raje, Bergamaschi

[4] Flow Control in a Resource-Sharing Computer Network - R. Kahn, W. Crowther

Web Links

1) http://www.topology-zoo.org/

2) https://arxiv.org/abs/2201.00484

3) https://www.hindawi.com/journals/wcmc/2018/7302025/

GitHub Repository Link

Thank You!