# Project Scope Report

**Collective Communication Operations on Mesh Topologies**

**Operations: Broadcast (Bcast) and Gather — Topologies: 2D and 3D Mesh**

**Aniket Gupta — Samarth Srikar**

## 1. Project Proposal

The goal of this project is to design, implement, and study how collective communication works on mesh-based networks, mainly focusing on **Broadcast** and **Gather** operations. Mesh topologies are common in many high-performance systems, and understanding how messages travel in such networks helps in improving performance and scalability.

In simple terms, our project will create working implementations of Bcast and Gather on both 2D and 3D mesh structures, run experiments, and observe how the communication cost changes when network dimensionality increases. We plan to measure time, communication steps, and message complexity based on the standard latency-bandwidth model. The final outcome will be a clear comparison showing how 3D mesh reduces delays compared to 2D mesh due to shorter communication paths.

The final report for the project will include code implementations, explanations of algorithms, runtime graphs, and conclusions about performance behavior.

## 2. Functionality of the Project

The project will support the following functionality:

- Implementing Broadcast (Bcast) on 2D and 3D mesh topologies.
- Implementing Gather on 2D and 3D mesh topologies.
- Simulation or execution of message transfers based on nearest-neighbor communication.
- Performance analysis using the latency-bandwidth model

$$T = t_s + t_w m$$

- Comparison of runtimes, communication rounds, and scalability.
- Visualization (if time allows) of the message flow on mesh grids.

This functionality gives us a full end-to-end view: from implementation to analysis to outcome.

## 3. Workplan

To complete the project properly and within the deadlines, we will follow a straightforward step-by-step workplan:

- First, build the basic mesh topology structures (2D and 3D).
- Then implement Broadcast for both topologies.
- After that, implement Gather for both topologies.
- Once the algorithms are ready, we will measure communication time and steps.
- Collect results, compare performance of 2D vs 3D mesh, and generate graphs.
- Finally, prepare the project report with explanations and results.

This plan keeps the project simple and ensures we finish everything on time.

## 4. Technologies to Be Used

We plan to use the following technologies during the project:

- **Python with mpi4py** for implementing the communication algorithms.
- **C/C++ with MPI (OpenMPI or MPICH)** if required for performance checks.
- **Matplotlib or similar tools** for plotting graphs.
- **Linux environment** for running MPI programs.

The main focus is on correctness and understanding communication behavior, so we will choose tools that help us prototype quickly.

## 5. Timeline

Since today is 8 November, we need to submit the project scope by 11 November. After that, we get around two weeks before our exams start on the 24th. So we will complete the main work in that time.

- **8–11 November**: Finalize scope, proposal, and project design.
- **12–17 November**: Implement mesh structures and Broadcast (2D and 3D).
- **18–22 November**: Implement Gather (2D and 3D) and run performance tests.

- **23 November**: Prepare graphs, comparisons, and final report draft.
- **Post 24 November (if needed)**: Minor refinements, but most work will be done earlier.

This timeline is tight but realistic, and we can comfortably finish before exams.

## 6. Technical Background and Problem Description

The project focuses on designing and analyzing efficient algorithms for collective communication operations on mesh networks. Collective operations enable multiple processes to exchange data in a structured way, and Broadcast and Gather are two of the most fundamental operations used in parallel computing.

**Broadcast (Bcast)** sends a message from one process to all others, and **Gather** collects data from all processes into the root. Mesh networks are widely used in HPC architectures, making these operations highly relevant.

We use the latency-bandwidth model

$$T_{msg} = t_s + t_w m$$

to measure communication cost. Important network metrics such as diameter, bisection width, and connectivity also affect performance and scalability.

## 7. Algorithm Design

### Broadcast on a 2D Mesh

The root broadcasts along its row, then each node forwards the message down its column.

```
Algorithm Mesh_Broadcast(root, M):
  1. Broadcast M along the root's row.
  2. Row nodes become column leaders.
  3. Each leader broadcasts M down its column.
```

Runtime:

$$T_{bcast}^{2D} = 2(\sqrt{p} - 1)t_s + (p - 1)t_w m$$

### Gather on a 2D Mesh

```
Algorithm Mesh_Gather(root):
  1. Rows gather data toward row leaders.
  2. Row leaders send data up the root column.
```

# 8. Topology Visualization

### 2D Mesh

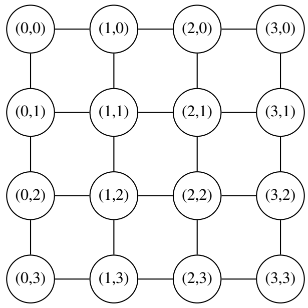


Figure 1: 2D Mesh (4x4 grid).

### 3D Mesh

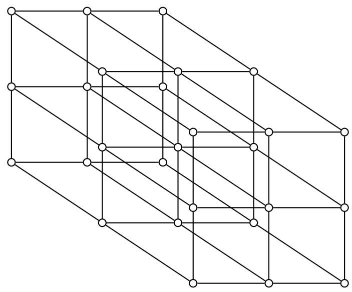

Figure 2: 3D Mesh (3x3x3 grid).

# 9. Expected Outcomes

- Working and tested Bcast and Gather algorithms for mesh networks.
- Detailed performance analysis for 2D vs 3D meshes.
- Insights on scalability and communication distance.
- A clear report summarizing algorithm behavior and results.