



NETFORMATION

CASTALIA PROJECT



CONTENTS

Problem Definition

Formation Protocol

Castalia Exploration

Network Simulation

PROBLEM DEFINITION

1.1. Understanding the Problem

- **Wireless sensor network (WSN):**
Provides Real-Time Monitoring, continuous collect data, allowing farmers to respond quickly to changing conditions.

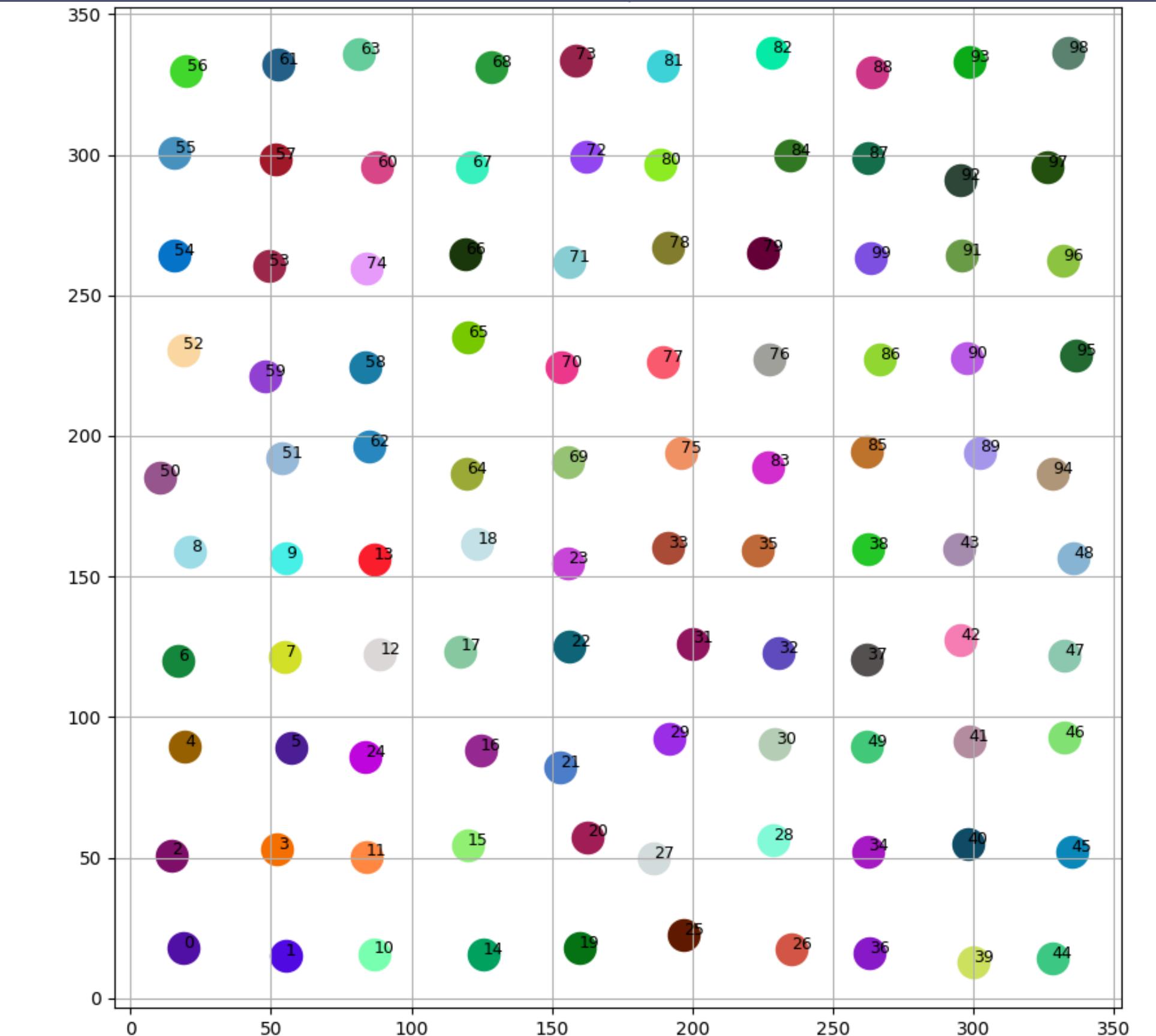


WSNs consist of spatially distributed sensor nodes that monitor various parameters, such as temperature, humidity, soil moisture and more

1.1. Understanding the Problem

Simulation network:

- Number of nodes: 100
- FoD: 400x400
- Base Station: node[0]
- Cluster head: 12, 56, 71, 90



1.2. Proposing a Solution

**DESIGN A ROUTING LAYER PROTOCOL TO MANAGE COMMUNICATION
BETWEEN SENSORS**

THE PROTOCOL CONSISTS OF 3 PHASES:

Phrase 1

Network Broadcasting

Phrase 2

Cluster Recruitment

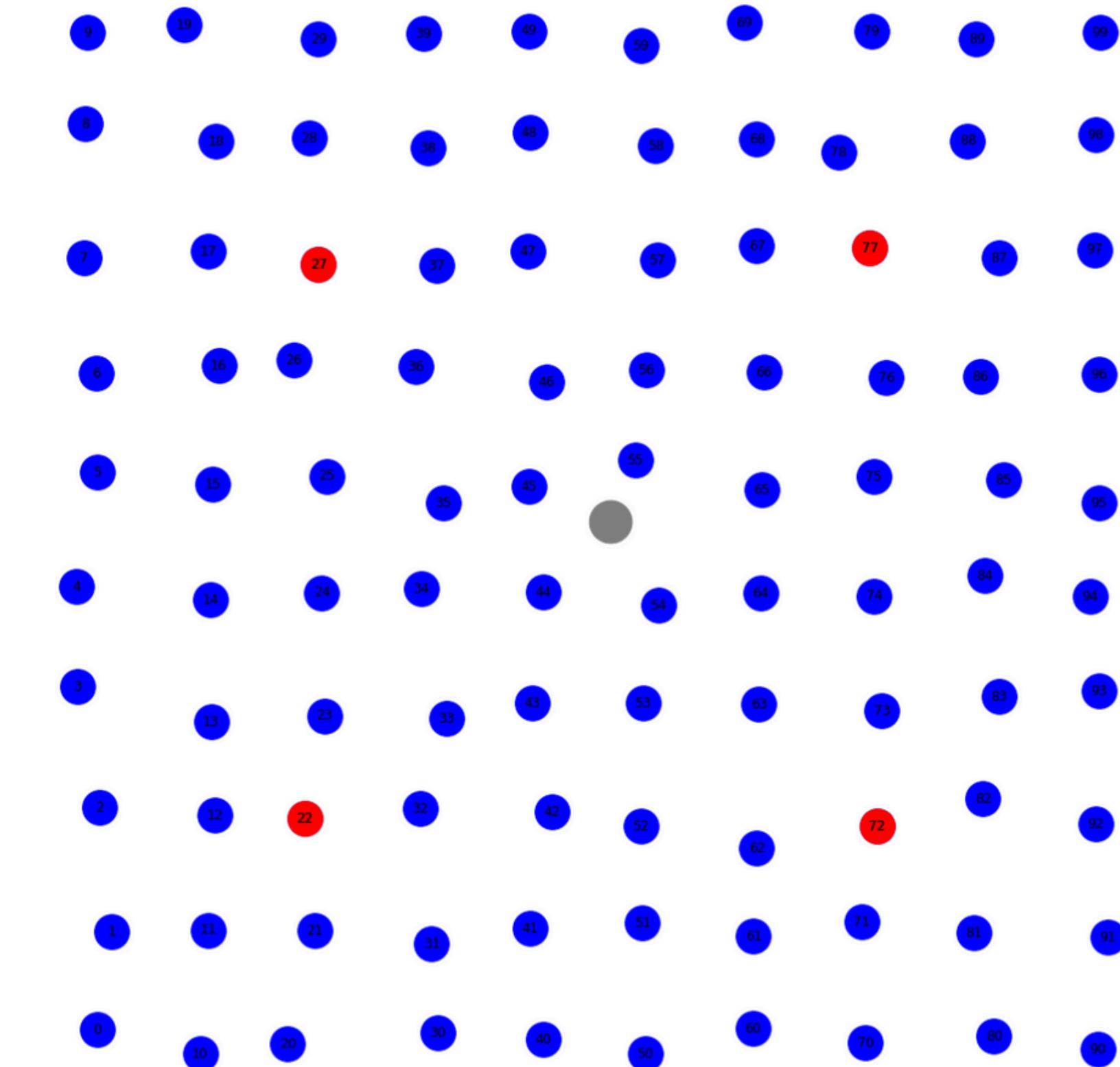
Phrase 3

Cluster Joining

1.2. Proposing a Solution

Phrase 1

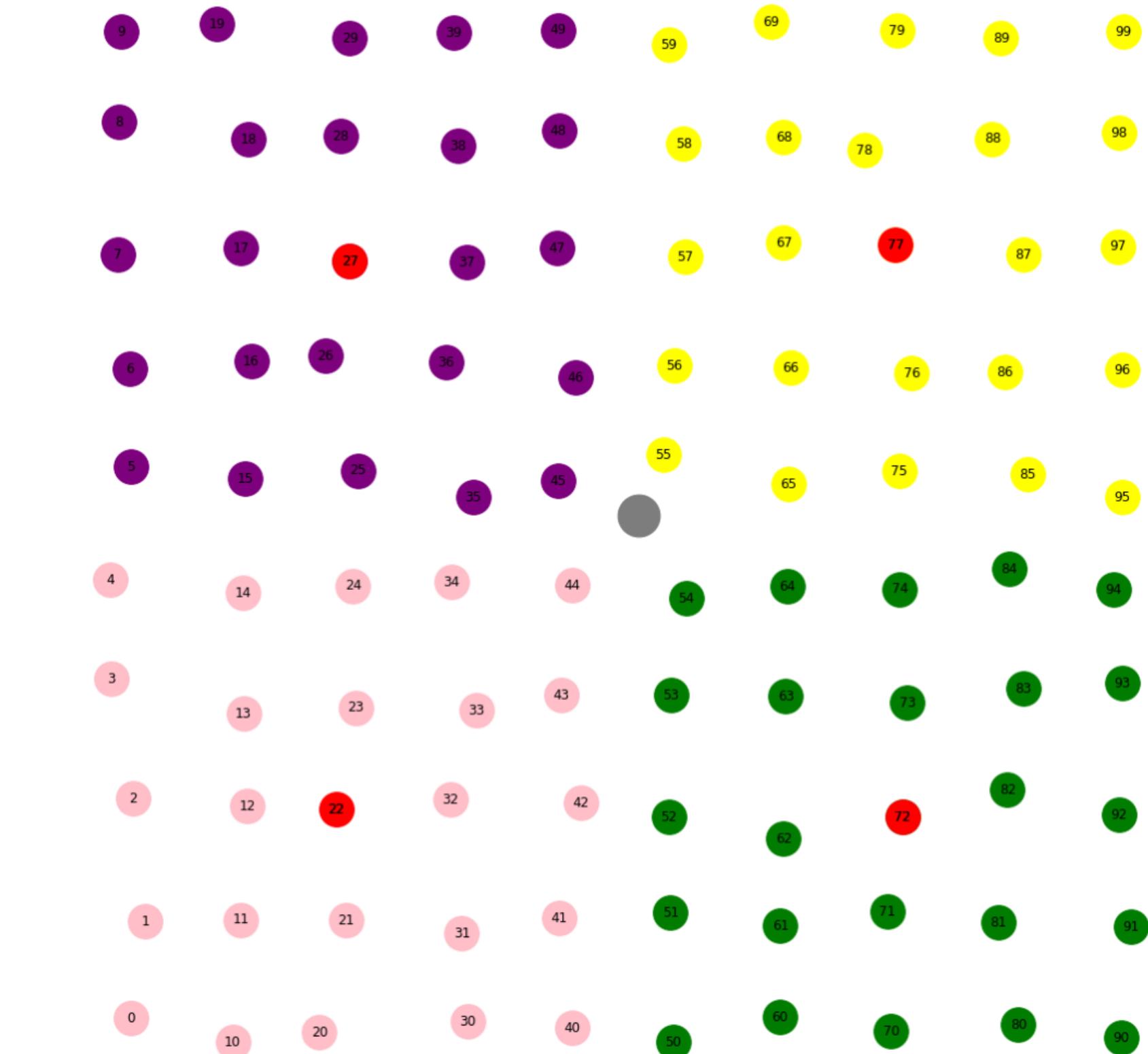
Network Broadcasting:
The Base Station broadcasts
a message across the entire
network to provide
information about
designated Cluster Heads



1.2. Proposing a Solution

Phrase 2

Cluster Recruitment:
Cluster Heads, after receiving the designation message, broadcast a recruitment message to nearby nodes.

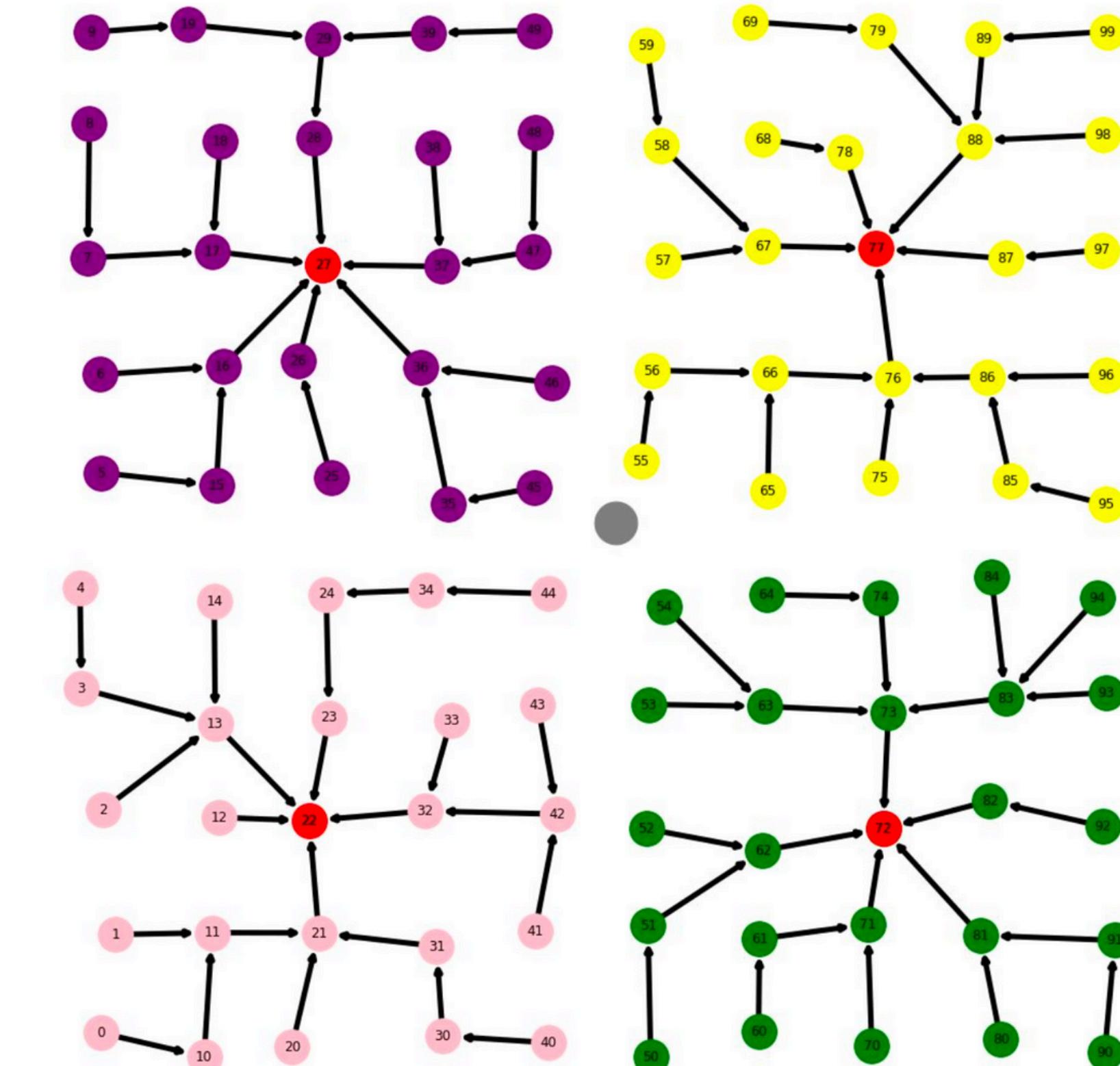


1.2. Proposing a Solution

Phrase 3

Cluster Joining:

Non-cluster-head nodes, upon receiving the first recruitment message, join the corresponding cluster and send a joining message back to the Cluster Head.



2. Protocol

PROTOCOL DESIGN

2. Protocol

CORRESPONDING TO THE THREE PHASES OF THE PROTOCOL ARE THREE TYPES OF PACKETS USED DURING THE NETWORK FORMATION PROCESS:

BCAST

Network Broadcasting Package

RECRUIT

Cluster Recruitment Packet

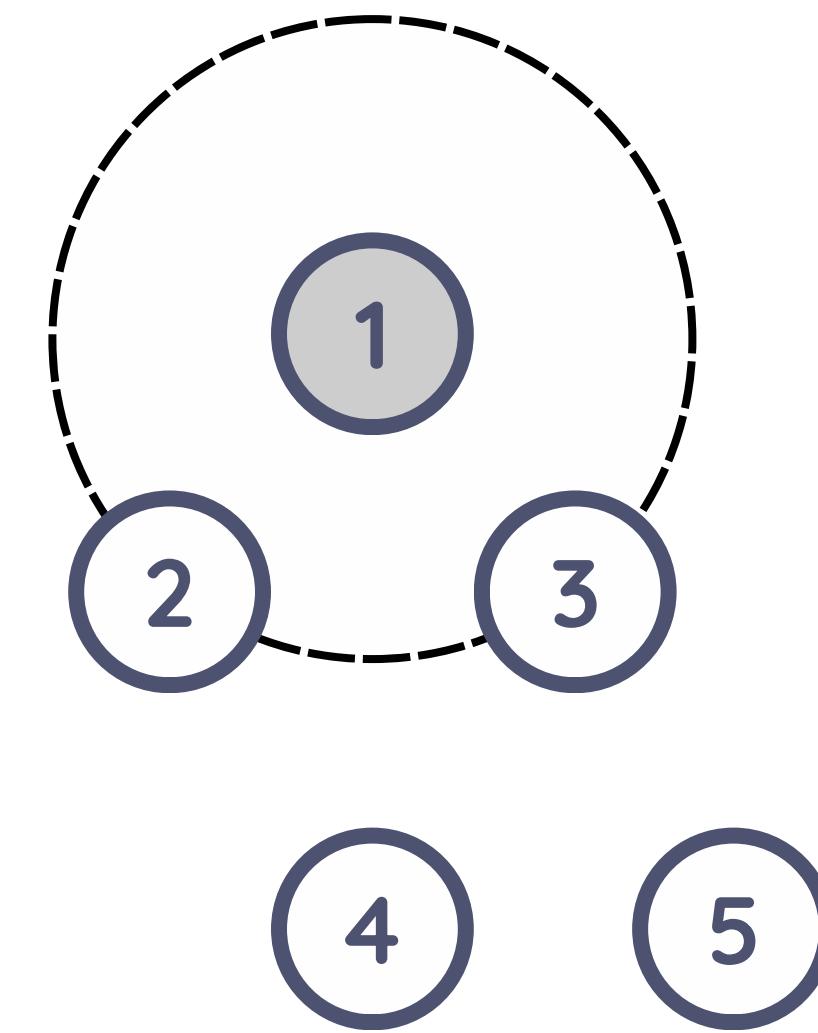
JOIN

Cluster Joining Packet

2.1. Broadcasting

Network Broadcasting:

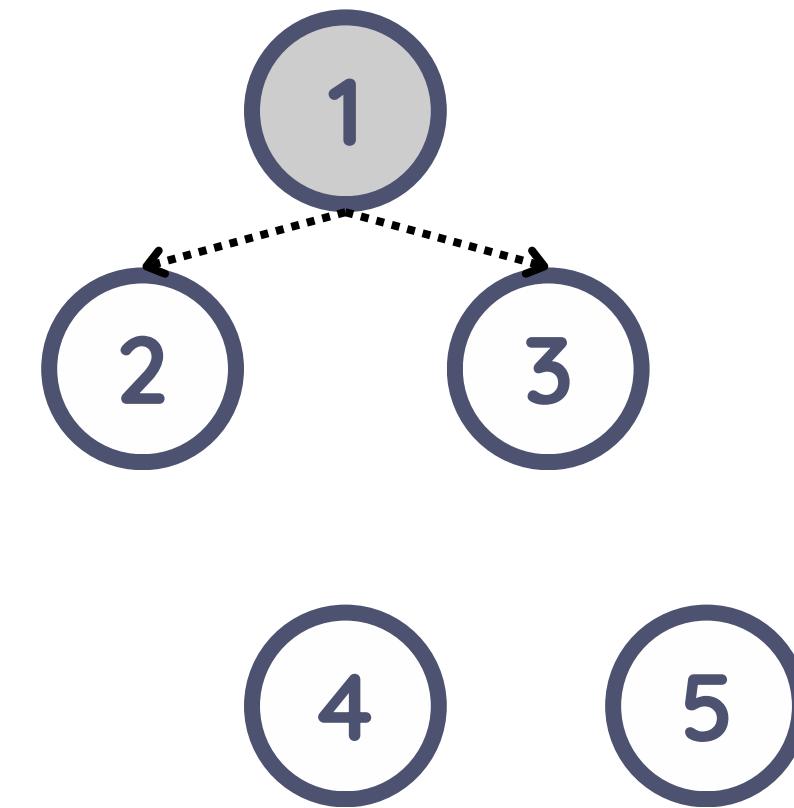
- At the start of Phase 1, the Base Station sends a BCAST packet to the nearest nodes.
- Other nodes, upon receiving the broadcast packet (if they have not already received a similar packet), process the information about Cluster Heads and forward the packet to their neighbors.
- This process continues until the packet reaches all nodes.



2.1. Broadcasting

Network Broadcasting:

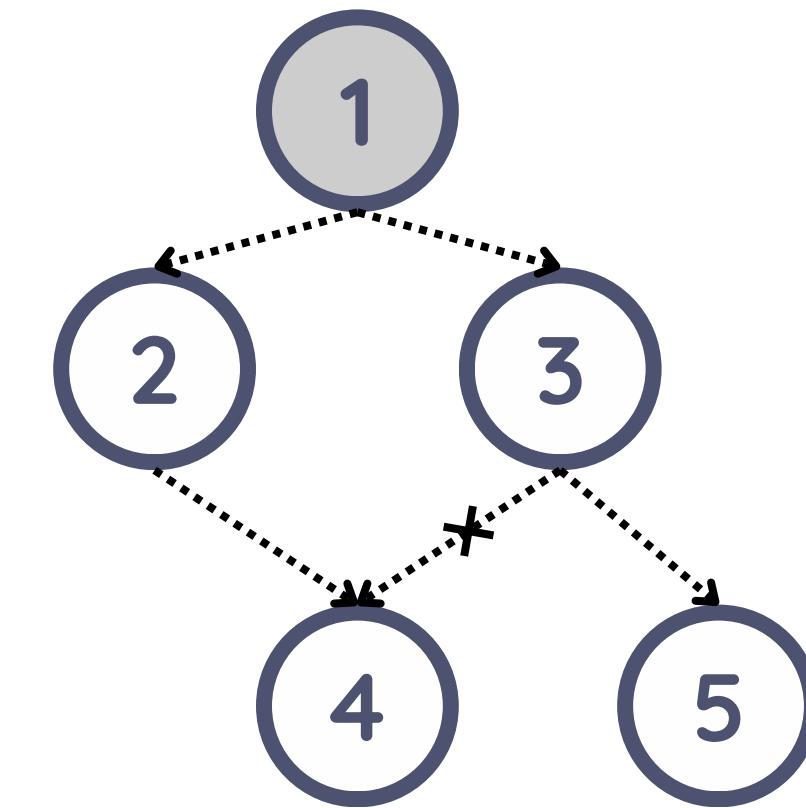
- At the start of Phase 1, the Base Station sends a BCAST packet to the nearest nodes.
- Other nodes, upon receiving the broadcast packet (if they have not already received a similar packet), process the information about Cluster Heads and forward the packet to their neighbors.
- This process continues until the packet reaches all nodes.



2.1. Broadcasting

Network Broadcasting:

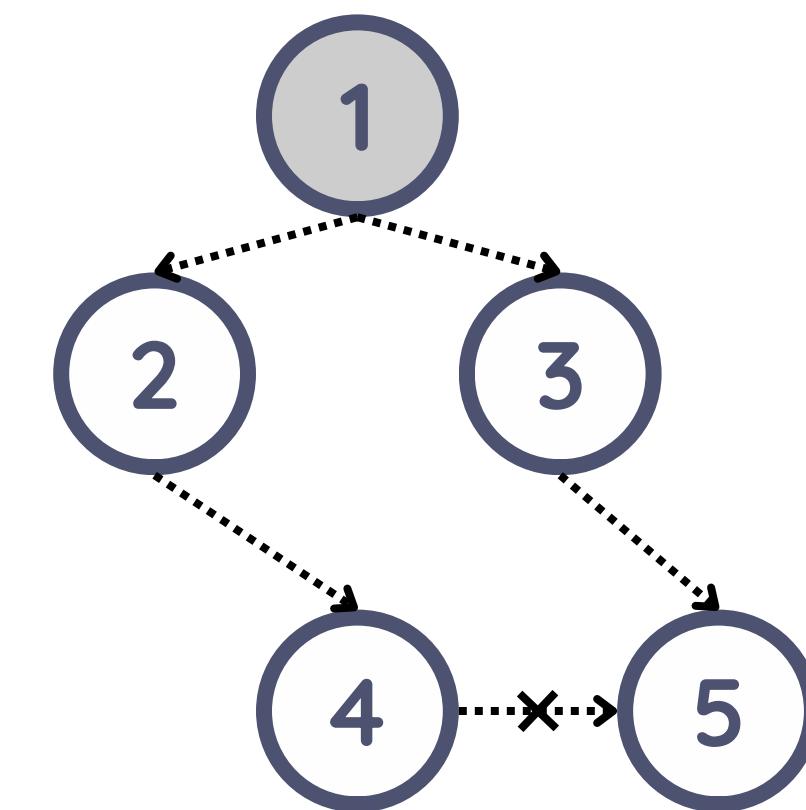
- At the start of Phase 1, the Base Station sends a BCAST packet to the nearest nodes.
- Other nodes, upon receiving the broadcast packet (if they have not already received a similar packet), process the information about Cluster Heads and forward the packet to their neighbors.
- This process continues until the packet reaches all nodes.



2.1. Broadcasting

Network Broadcasting:

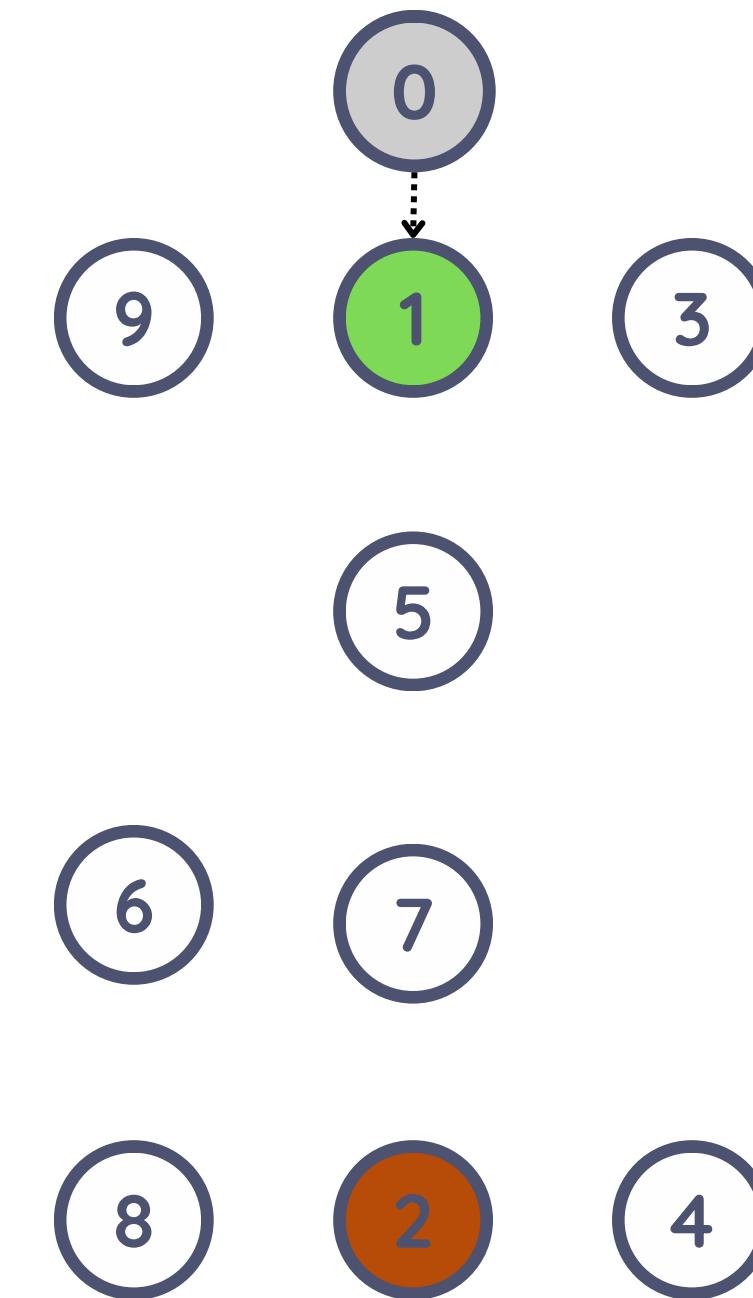
- At the start of Phase 1, the Base Station sends a BCAST packet to the nearest nodes.
- Other nodes, upon receiving the broadcast packet (if they have not already received a similar packet), process the information about Cluster Heads and forward the packet to their neighbors.
- This process continues until the packet reaches all nodes.



2.2. Cluster Recruitment

Cluster Recruitment:

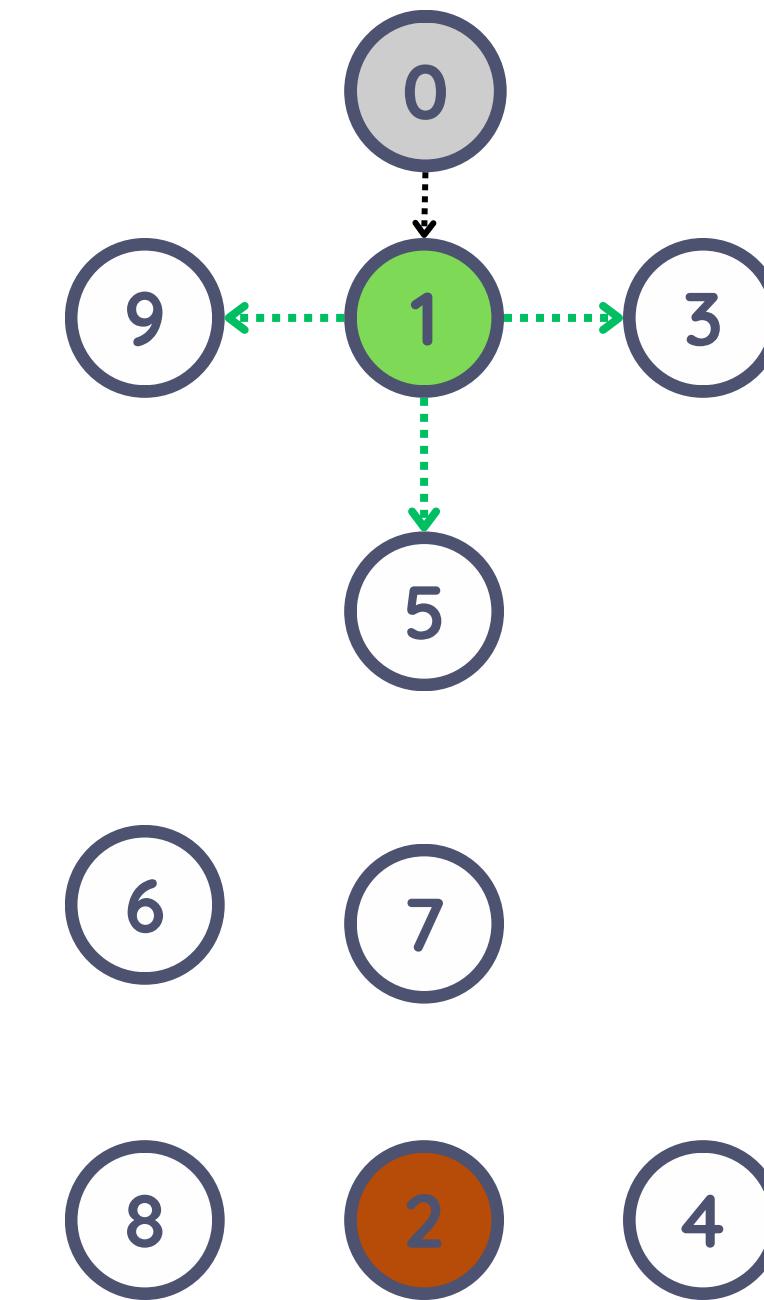
- When a node is designated as a Cluster Head (after receiving a BCAST packet), it sends a RECRUIT packet to nearby nodes.
- Other nodes, upon receiving the RECRUIT packet, join the cluster (if they have not already joined another cluster) and forward the RECRUIT packet to other nodes.
- If a node that has already joined a cluster receives a recruitment message, it will automatically ignore the message.



2.2. Cluster Recruitment

Cluster Recruitment:

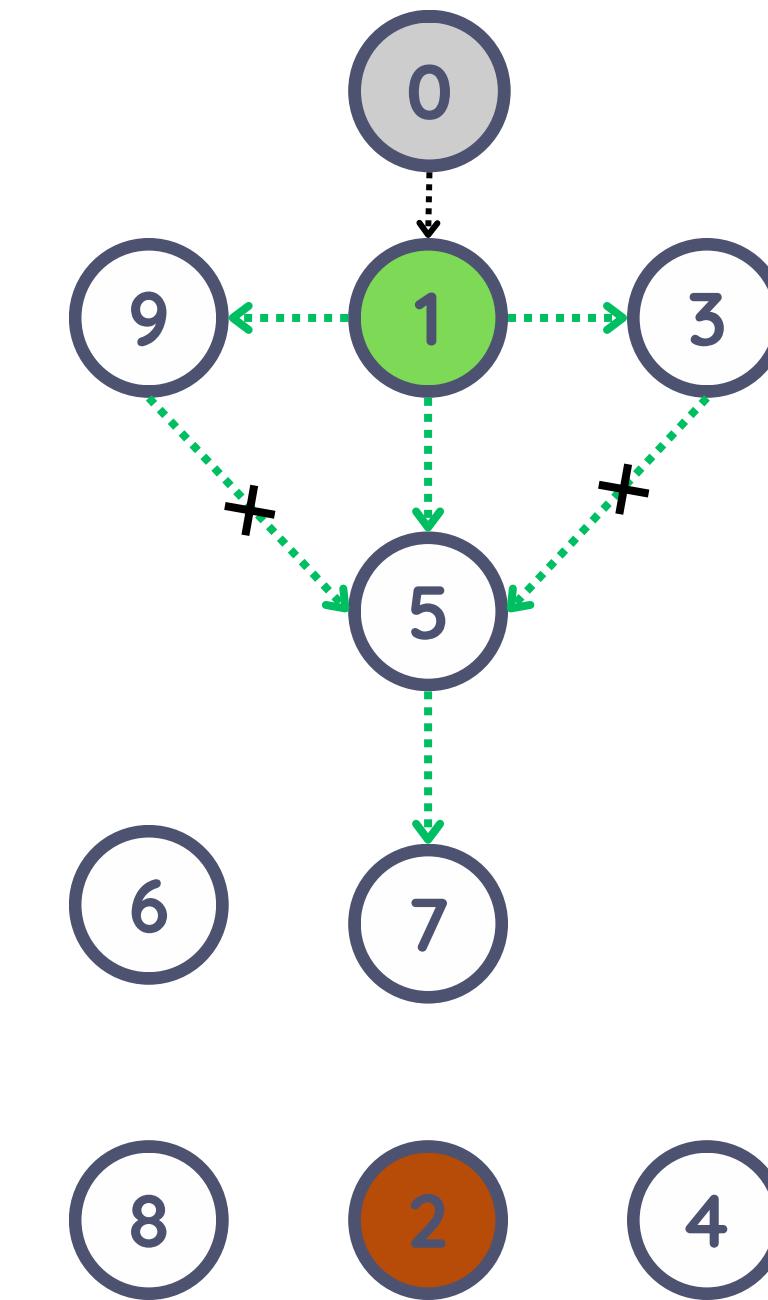
- When a node is designated as a Cluster Head (after receiving a BCAST packet), it sends a RECRUIT packet to nearby nodes.
- Other nodes, upon receiving the RECRUIT packet, join the cluster (if they have not already joined another cluster) and forward the RECRUIT packet to other nodes.
- If a node that has already joined a cluster receives a recruitment message, it will automatically ignore the message.



2.2. Cluster Recruitment

Cluster Recruitment:

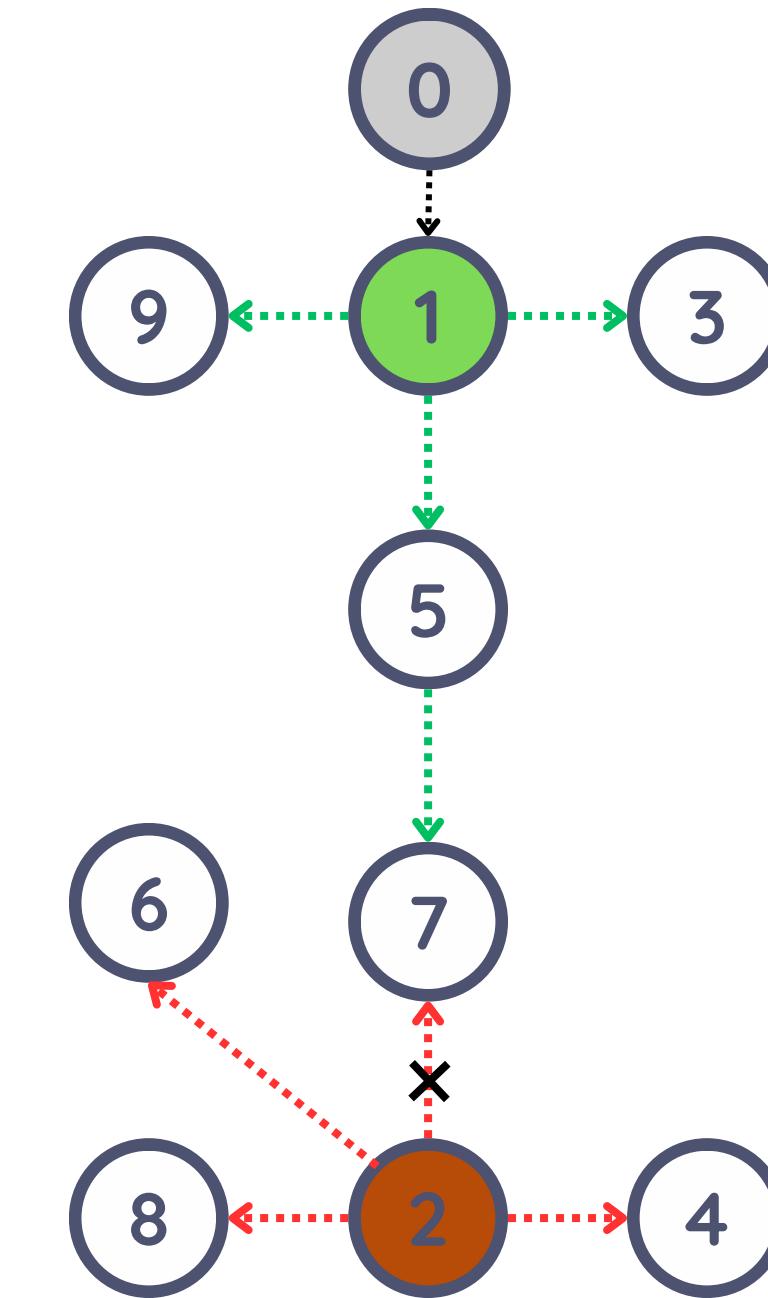
- When a node is designated as a Cluster Head (after receiving a BCAST packet), it sends a RECRUIT packet to nearby nodes.
- Other nodes, upon receiving the RECRUIT packet, join the cluster (if they have not already joined another cluster) and forward the RECRUIT packet to other nodes.
- If a node that has already joined a cluster receives a recruitment message, it will automatically ignore the message.



2.2. Cluster Recruitment

Cluster Recruitment:

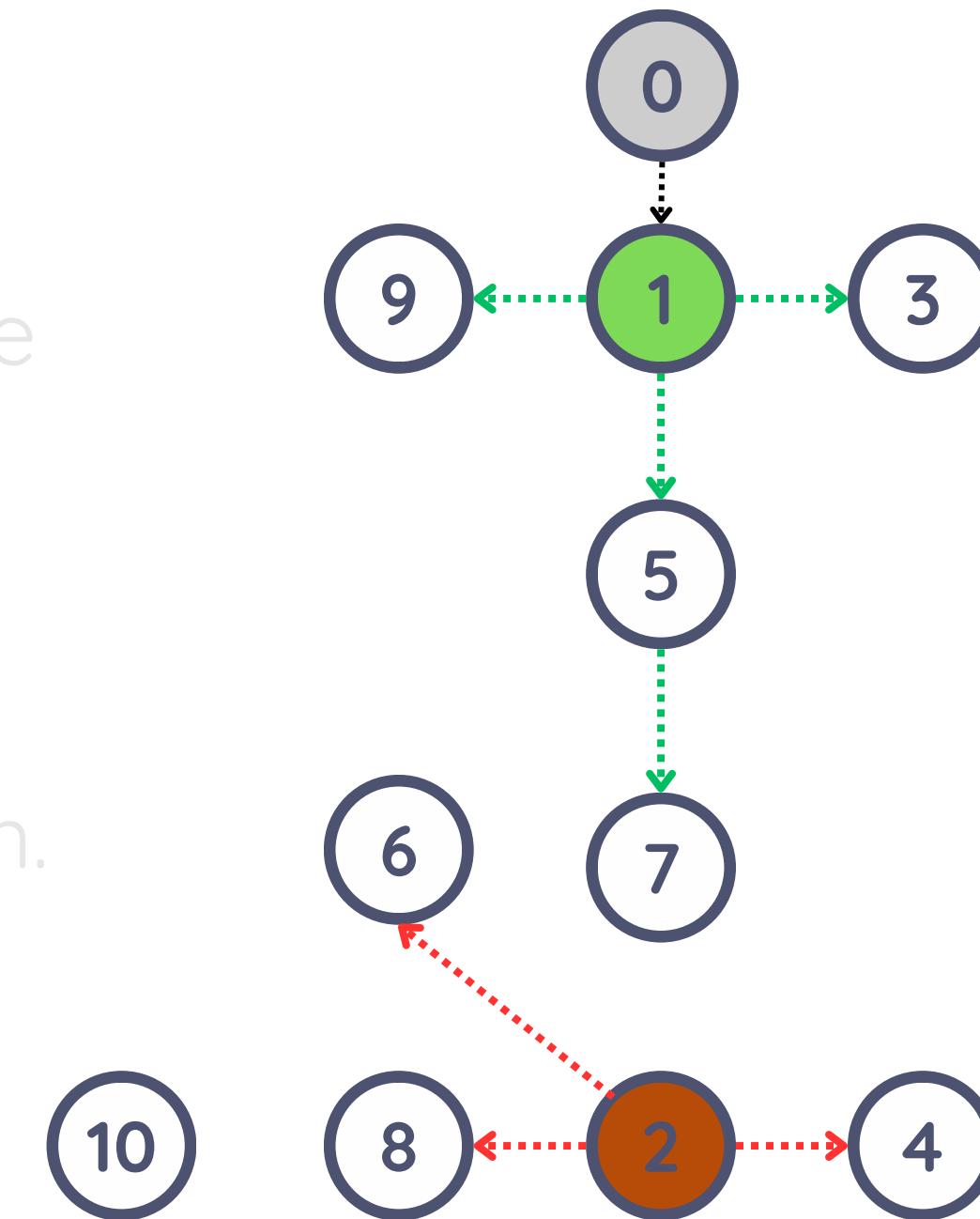
- When a node is designated as a Cluster Head (after receiving a BCAST packet), it sends a RECRUIT packet to nearby nodes.
- Other nodes, upon receiving the RECRUIT packet, join the cluster (if they have not already joined another cluster) and forward the RECRUIT packet to other nodes.
- If a node that has already joined a cluster receives a recruitment message, it will automatically ignore the message.



2.3. Joining Cluster

Joining Cluster:

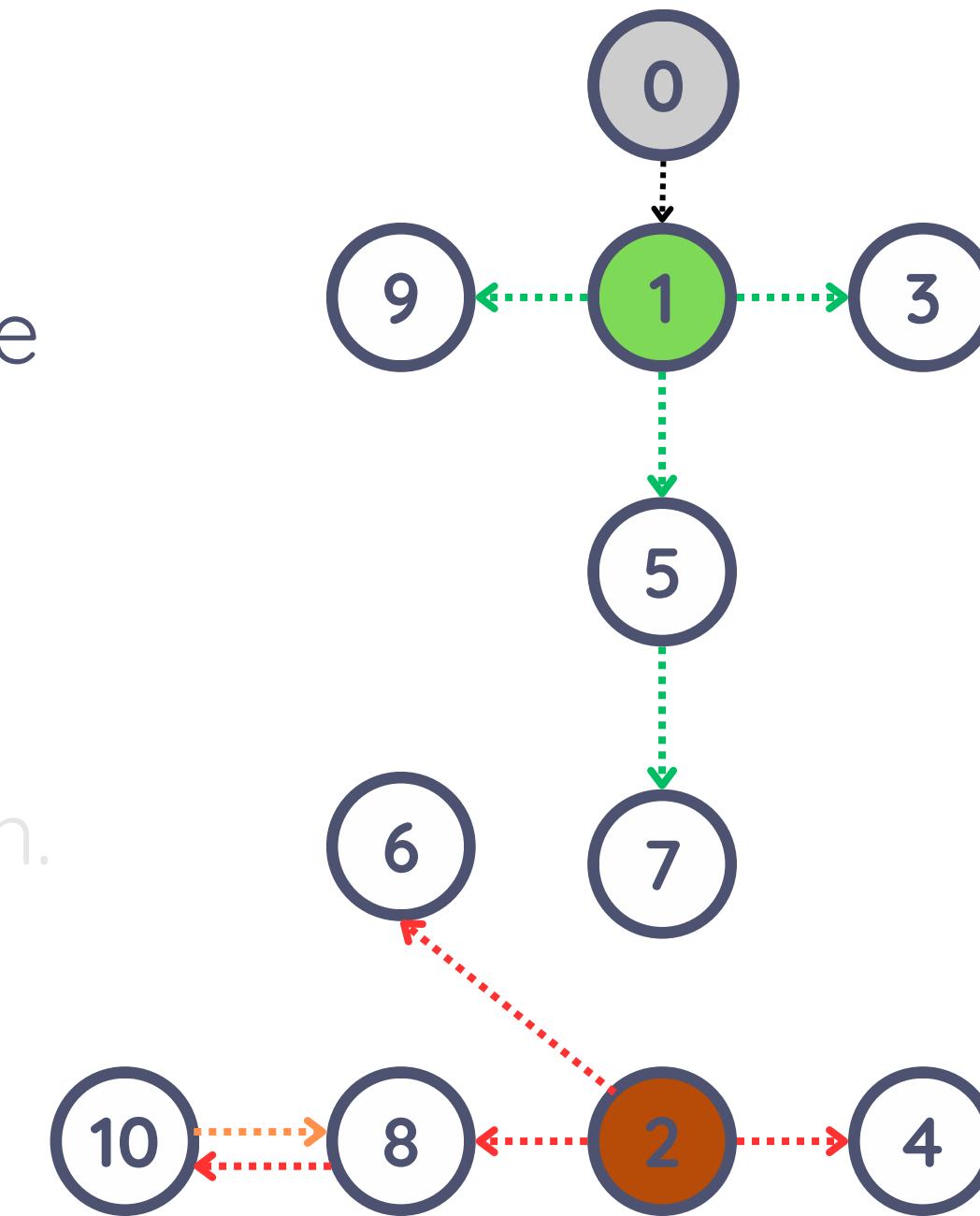
- When a non-cluster-head node joins a cluster, it sends a JOIN packet back to the previous node.
- If the packet has not yet reached the Cluster Head, it is forwarded until the Cluster Head receives the join information.



2.3. Joining Cluster

Joining Cluster:

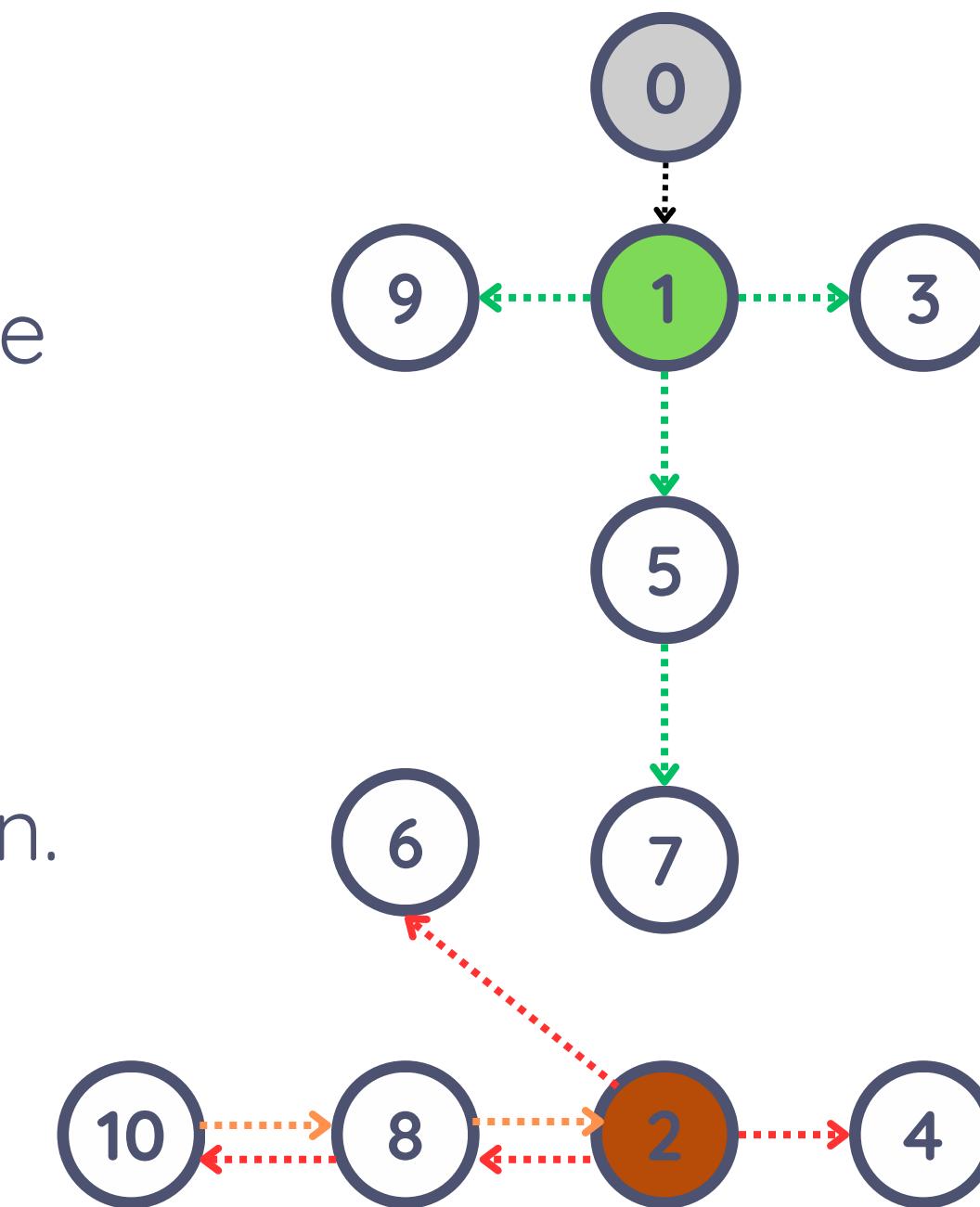
- When a non-cluster-head node joins a cluster, it sends a JOIN packet back to the previous node.
 - If the packet has not yet reached the Cluster Head, it is forwarded until the Cluster Head receives the join information



2.3. Joining Cluster

Joining Cluster:

- When a non-cluster-head node joins a cluster, it sends a JOIN packet back to the previous node.
 - If the packet has not yet reached the Cluster Head, it is forwarded until the Cluster Head receives the join information



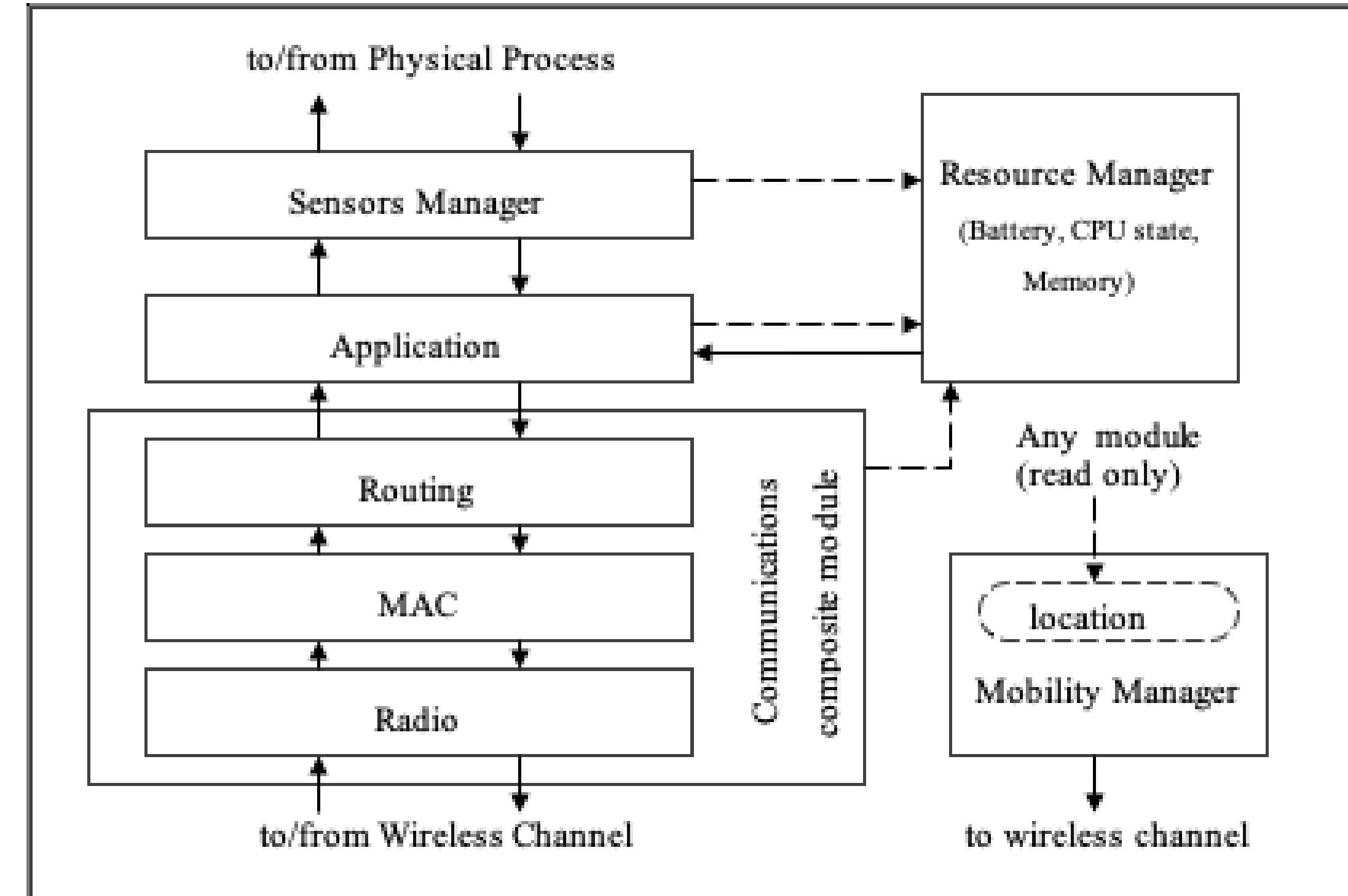
3. Castalia

CASTALIA



3.1. Overview

Each node is a composite module with submodules for resource management, application logic, MAC protocols, routing, and mobility.



3.2. Deployment Plan

IMPLEMENT THE PROTOCOL USING A ROUTING LAYER MODULE IN CASTALIA

NED file

Defines the module

MSG file

Defines protocol packets

.cc, .h files

Defines protocol behaviour

3.3. Protocol program

simple WSNRouting like node.communication.routing.iRouting

```
{  
parameters:  
    bool collectTraceInfo = default (false);  
    int maxNetFrameSize = default (0); // bytes  
    int netDataFrameOverhead = default (10); // bytes  
    int netBufferSize = default (32); // number of messages  
  
gates:  
    output toCommunicationModule;  
    output toMacModule;  
    input fromCommunicationModule;  
    input fromMacModule;  
    input fromCommModuleResourceMgr;  
}
```

**DEFINING
PROTOCOL
MODULE**

3.3. Protocol program

```
enum WSNRoutingPacket_Type
{
    BCAST = 1;
    ACK = 2;
    RECRUIT = 3;
    JOIN = 4;
};
```

```
packet WSNRoutingPacket extends RoutingPacket {
    int WSNRoutingMessage enum (WSNRoutingPacket_Type);
    int messageID;
    string clusterAdd[10];
    string origin;
    string source;
    string destination;
}
```

DEFINING PROTOCOL PACKETS

3.3. Protocol program

```
class WSNRouting: public VirtualRouting {  
private:  
    bool isSink;  
    std::string cluster[10];// cluster head temp array  
    std::string origin;  
    std::string ch = "-1"; // cluster head id  
    int lastPkt = -1; // last received packet id  
    std::string bHop; // the previous hop id  
  
    std::vector<std::string> clusterUnits;  
    std::map<std::string, std::string> routingTable;
```

```
protected:  
    void startup();  
    void fromApplicationLayer(cPacket *, const char *);  
    void fromMacLayer(cPacket *, int, double, double);  
    //void finish();  
};
```

**DEFINING
PROTOCOL
ATTRIBUTES**

3.3. Protocol program

PROTOCOL STARTUP BEHAVIOURS

```
Define_Module(WSNRouting);

void WSNRouting::startup(){
    trace() << "Startup " << SELF_NETWORK_ADDRESS << " node ";
    cModule *appModule = getParentModule()->getParentModule()->getSubmodule("Application");
    if (appModule->hasPar("isSink"))
        isSink = appModule->par("isSink");
    else
        throw cRuntimeError("\nThis application has no parameter isSink");
    if (isSink){
        // broadcast message
    }
}
```

3.3. Protocol program

```
void WSNRouting::fromMacLayer(cPacket * pkt, int srcMacAddress, double rssi, double lqi)
{
    // Cast the packet
    WSNRoutingPacket *netPacket = dynamic_cast <WSNRoutingPacket*>(pkt);
    if (!netPacket) {
        return;
    }

    //Get traces
    int pkt_kind = netPacket->getWSNRoutingMessage();
    const char * org = netPacket->getOrigin();
    const char * from = netPacket->getSource();

    switch (pkt_kind) {
    case WSNRoutingPacket_Type::BCAST: { //Broadcast packet processing }
    case WSNRoutingPacket_Type::RECRUIT: { //Recruit packet processing }
    case WSNRoutingPacket_Type::JOIN: { // Join packet processing }
    }
}
```

PROCESS
PROTOCOL
PACKET

4. Simulation

PROTOCOL SIMULATION

4.1. Run simulation

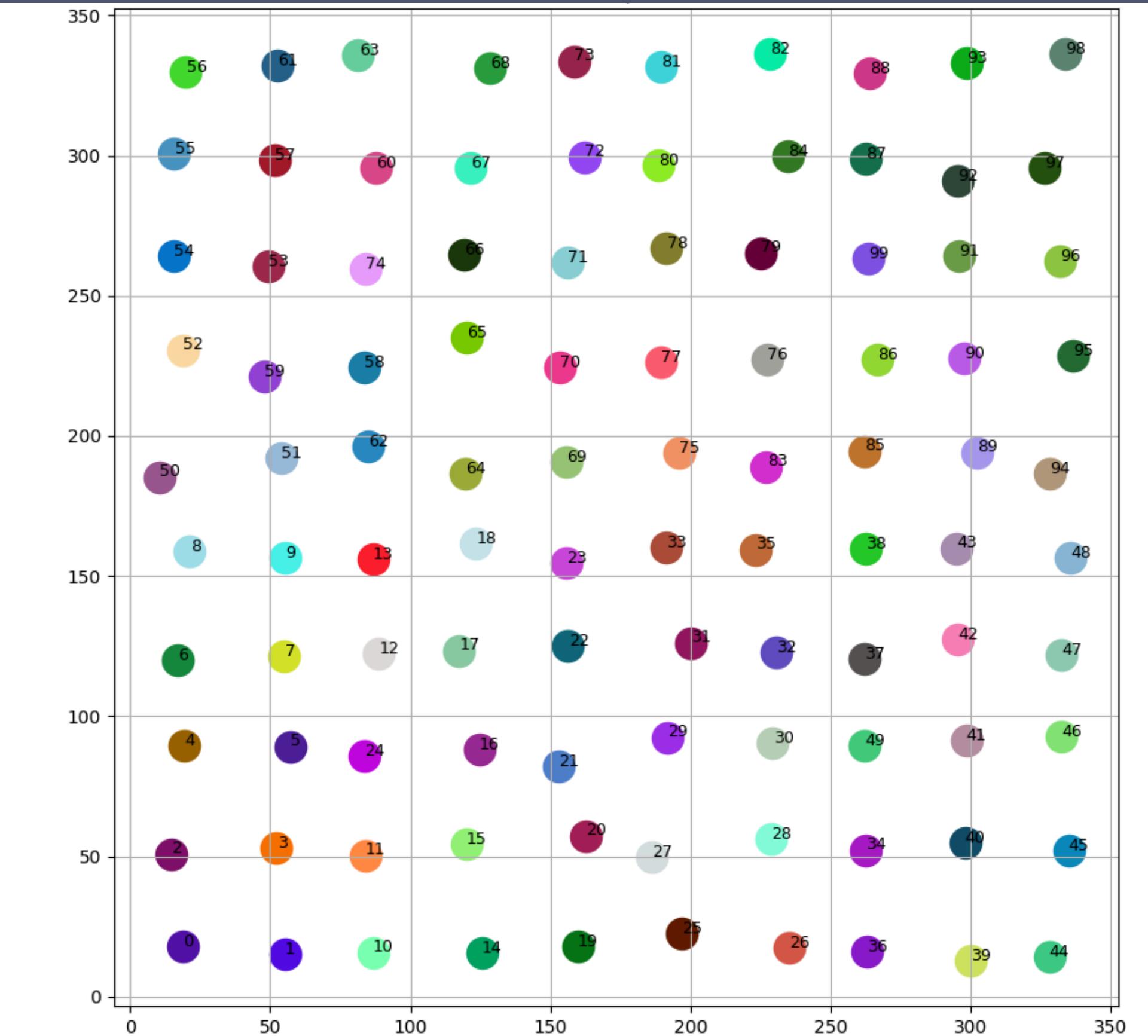
Run Simulation

Sensor numbers: 100

Base Station: node 0

Simulation time: 1000

Cluster heads: 12, 56, 71, 90



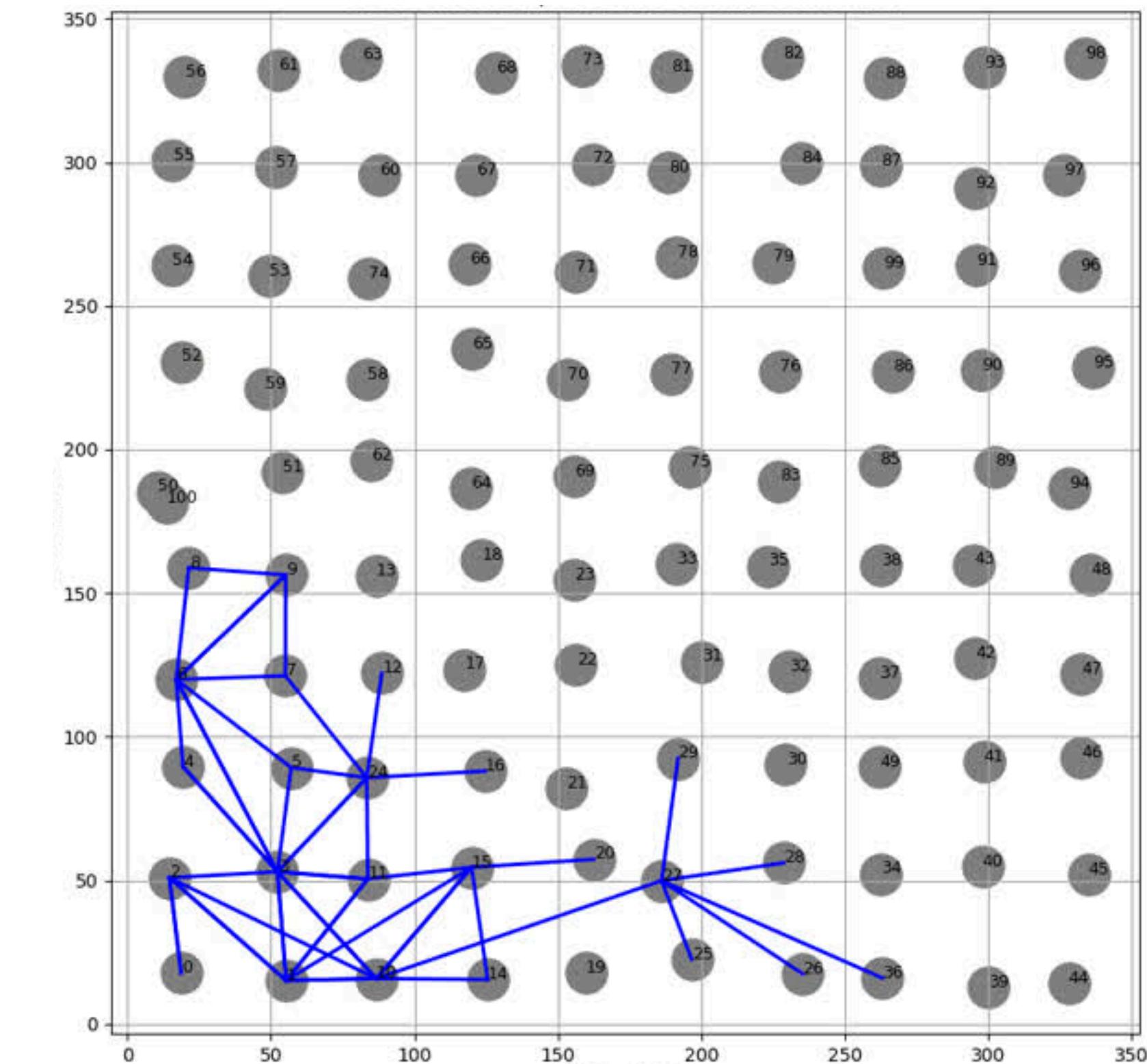
4.1. Run simulation

Phrase 1: Broadcasting

Packets sent: 338

Sim time: 10.5

Involved node: 99/100



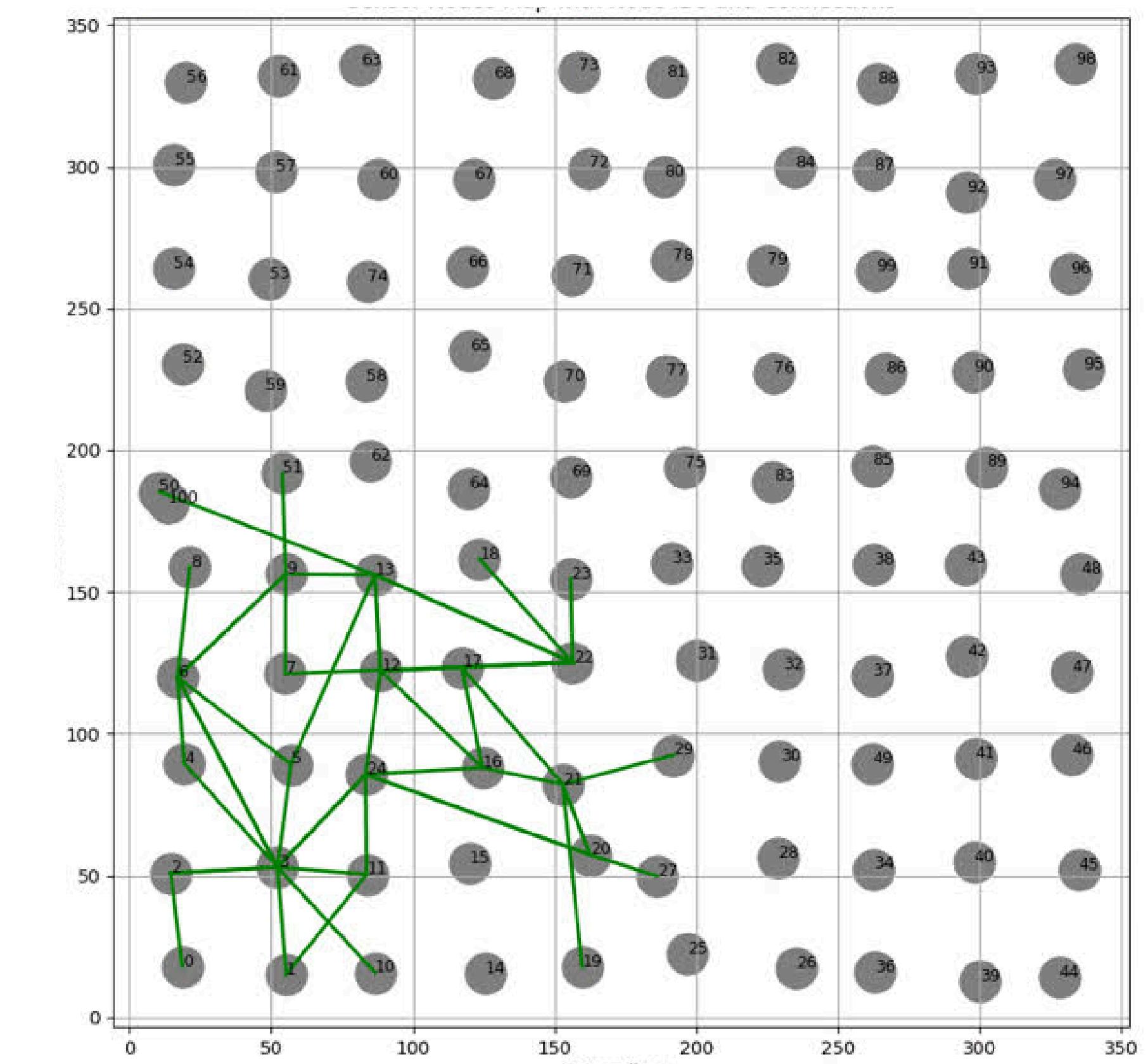
4.1. Run simulation

Phrase 2: Cluster recruiting

Packets sent: 363

Sim time: 12.34

Involved node: 96/100



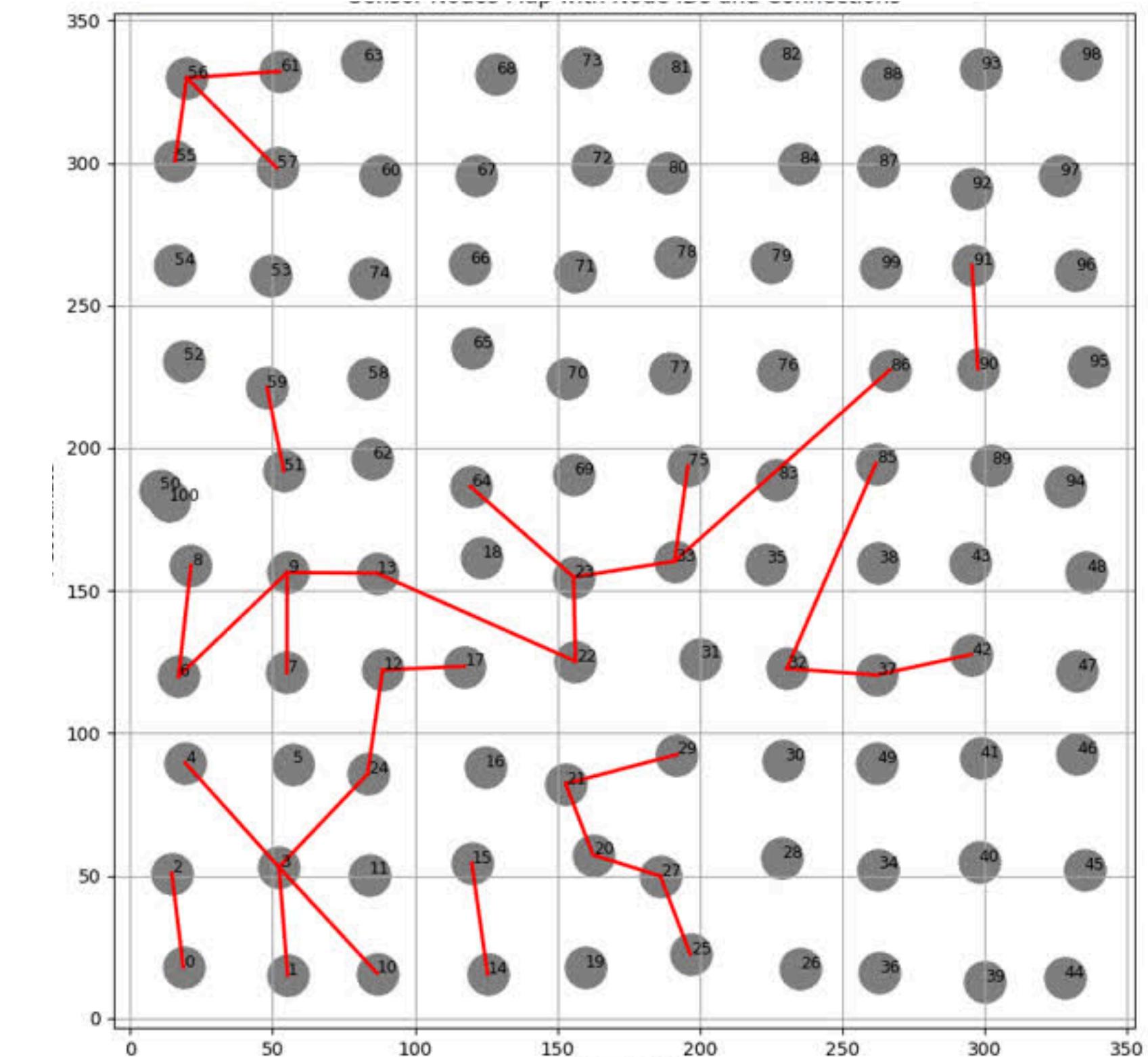
4.1. Run simulation

Phrase 3: Cluster Joining

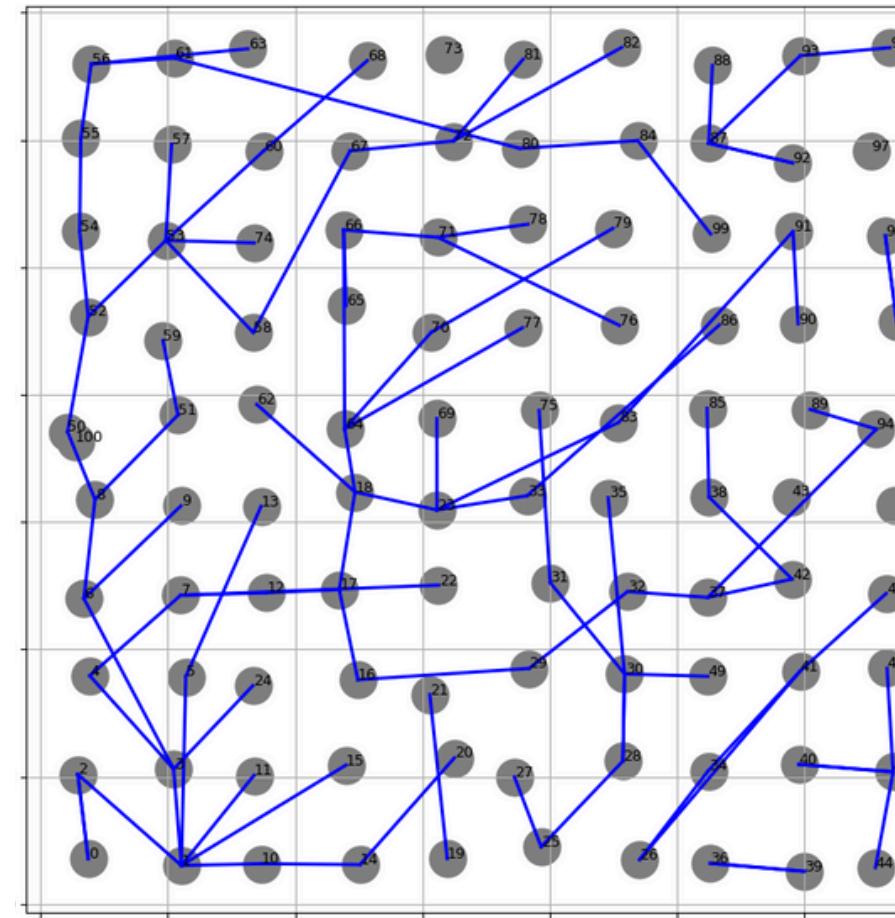
Packets sent: 60

Sim time: 12.33

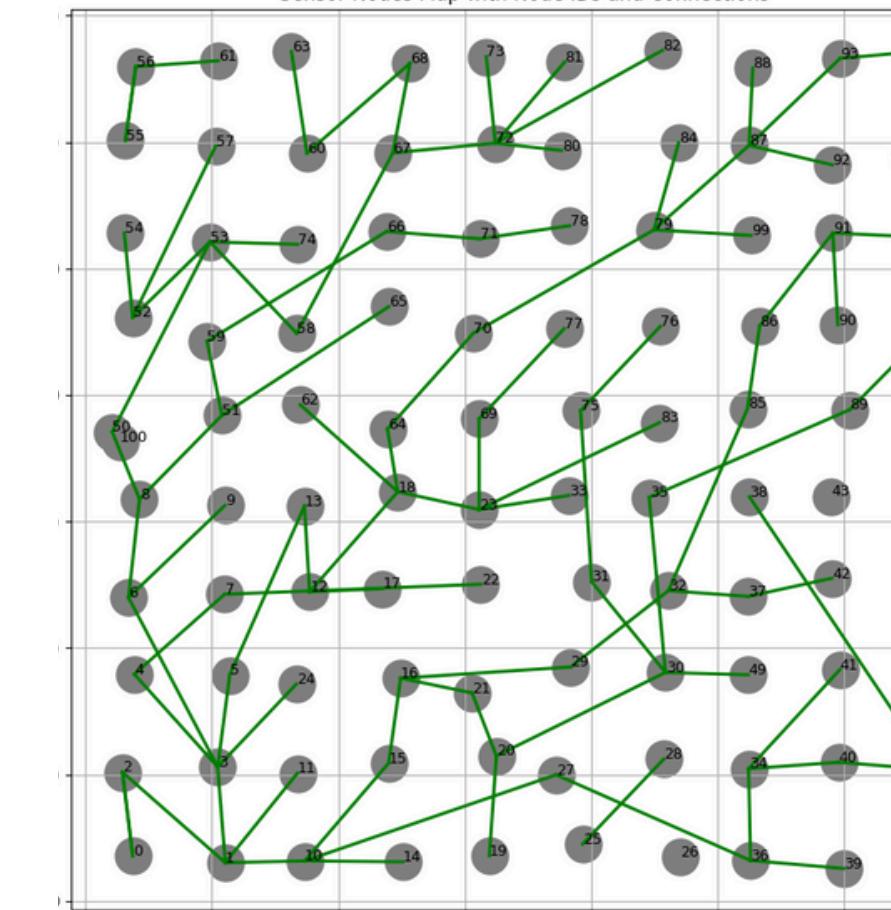
Involved node: 78/100



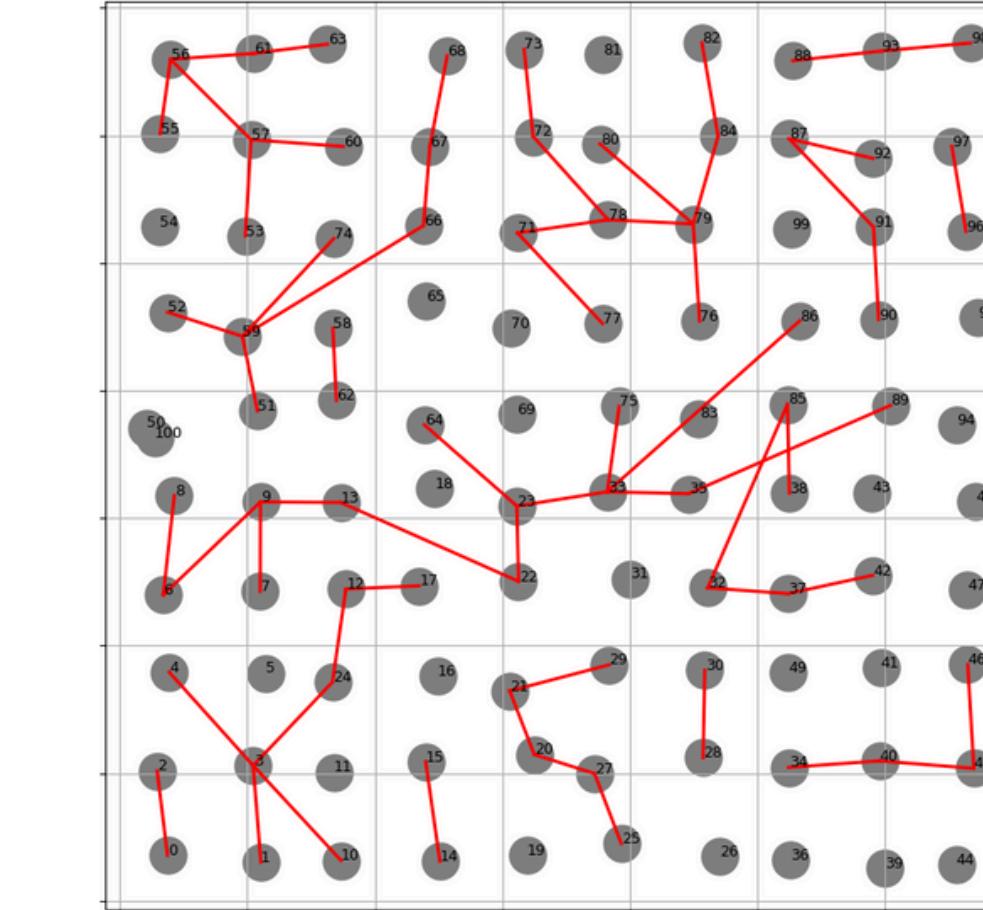
4.2. Solution Evaluation



Phrase 1
99%



Phrase 2
96%



Phrase 3
78%

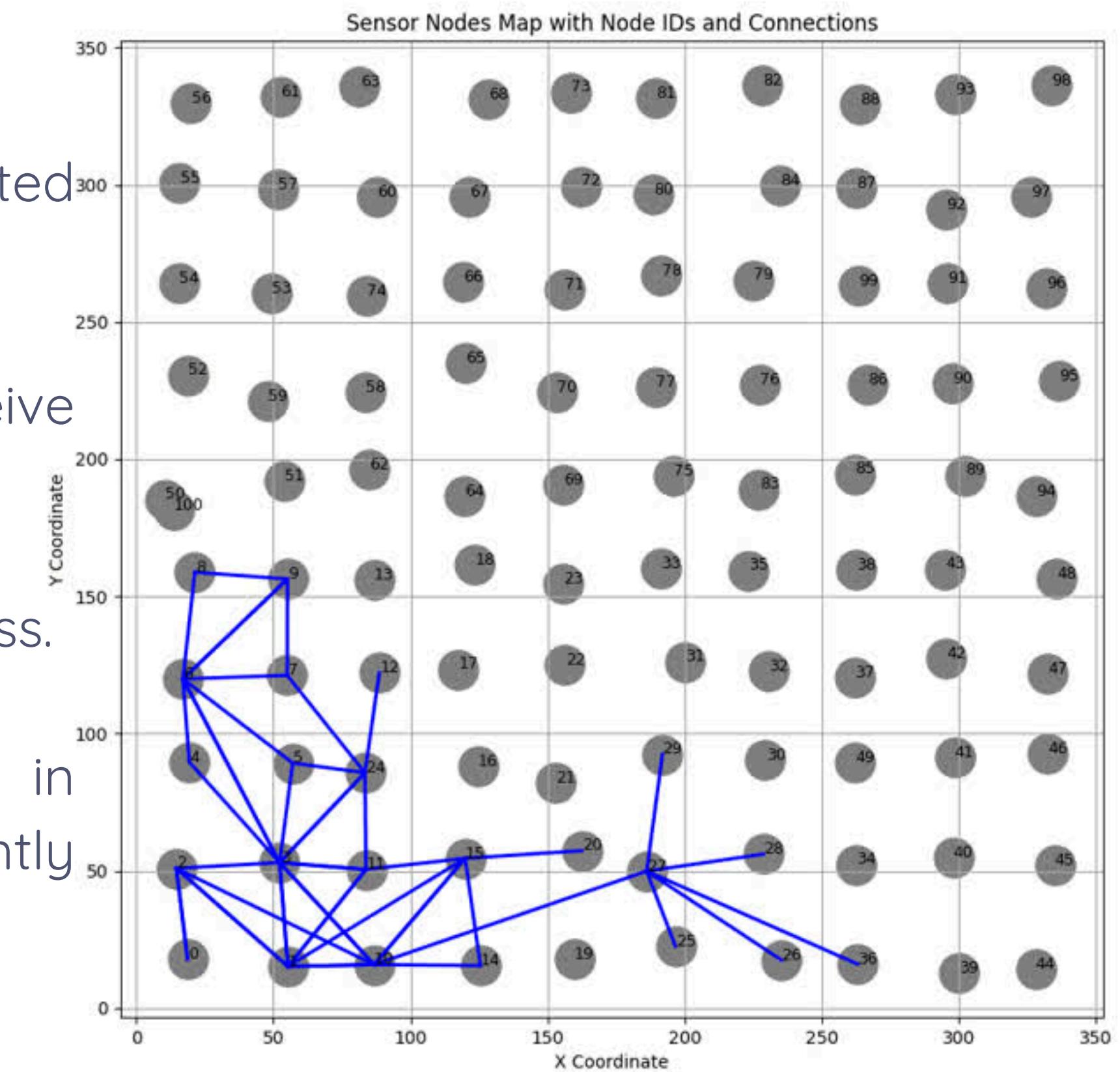
4.3. Conclusion

Strengths:

- Minimizes the number of transmitted packets
- Relatively fast execution time.
- A large number of nodes receive broadcast messages.

Limitations:

- Packets are prone to collisions and loss.
- Routing is not optimized.
- The number of nodes participating in the recruitment process is significantly reduced.



THANKS FOR LISTENING

CASTALIA PROJECT