Parameters for HEACT (from your scenario):

- NETWORK_SIZE = 100 x 100
- BS_POS = (50, 150)
- NUM_NODES = 10
- INITIAL_ENERGY = 0.5 J
- HEACT_P_CH = 0.1 (used by select_heact_chs_refined with HEACT_DIST_FACTOR_CH_SELECTION)
- HEACT_MIN_ENERGY_FOR_CH_CANDIDACY = 0.1 J
- HEACT_MAX_CLUSTER_SIZE = 15
- HEACT_DIST_FACTOR_CH_SELECTION = 1.1
- HEACT_INITIAL_DIRECT_INTERVALS = 1
- HEACT_RECLUSTER_INTERVAL = 3
- HEACT_MIN_ENERGY_FOR_CH_RELAY = 0.2 J
- HEACT_TREE_ENERGY_PENALTY_EXPONENT = 1.5
- Initial Energy = 0.5 J per node.

See Illustration 1 for the nodes distribution .

Dry Run - Round 1 (Initial Direct Phase)

1. Call to run_simulation_protocol (HEACT block):

- round_num = 1
- heact_reconfig_count (initially 0)
- Condition (round_num - 1) % HEACT_RECLUSTER_INTERVAL == 0 or round_num == 1 is TRUE.
- heact_reconfig_count becomes 1.
- is_initial_phase = (1 <= 1) which is TRUE.
- build_tree_this_round = False.

2. Call to select_heact_chs (nodes, HEACT_P_CH, HEACT_MIN_ENERGY_FOR_CH_CANDIDACY, HEACT_DIST_FACTOR_CH_SELECTION):

- **Input:** nodes (all 10 nodes, E=0.5J each), p_ch=0.1, min_energy_threshold=0.1J, dist_factor=1.1.
- **Internal Steps:**
    - alive_nodes = all 10 nodes.
    - total_energy_alive = 10 * 0.5 = 5.0 J.

- avg_dist_bs: Let's assume this is calculated to be 90m.
- **For each node n:**
  - n.reset_heact_round_state() is called.
  - n.energy (0.5) >= min_energy_threshold (0.1) is TRUE for all.
  - base_prob = 0.1 * (10 * 0.5 / 5.0) = 0.1.
  - distance_modifier calculated:
    - For N0 (Far, dist ~146m, rel_dist ~1.62): modifier ≈ 1.0 + (0.1 * min(1, 0.62/1)) ≈ 1.062
    - For N8 (Close, dist ~28m, rel_dist ~0.31): modifier ≈ (1/1.1) + (1-1/1.1)*(1-0.69) ≈ 0.909 + 0.091*0.31 ≈ 0.937
  - threshold(N0) ≈ 0.1 * 1.062 = 0.1062
  - threshold(N8) ≈ 0.1 * 0.937 = 0.0937
  - Random numbers generated.
- **Hypothetical Outcome:** Assume random_N4 < threshold(N4) and random_N8 < threshold(N8).

- **Output:** heact_cluster_heads = [N4_object, N8_object], setup_energy_select = {N0:0, N1:0,...} (as no energy explicitly deducted *in* this function in the code). ***(Illustration 2)***

3. Call to form_heact_clusters(nodes, heact_cluster_heads, HEACT_MAX_CLUSTER_SIZE):

- **Input:** nodes, heact_cluster_heads = [N4, N8], max_cluster_size = 15.
- **Internal Steps:**
  - alive_members = {N0, N1, N2, N3, N5, N6, N7, N9}. ch_map = {N4.id: N4, N8.id: N8}.
  - clusters = {N4.id: [], N8.id: []}, current_cluster_sizes = {N4.id:0, N8.id:0}.
  - Members are shuffled. Let's take N0 first:
    - N0: min_dist_sq found with N4. best_ch_id = N4.id.
    - dist_to_ch(N0, N4) ≈ 94m.
    - e_tx_join(N0) = calculate_transmit_energy(200, 94) (uses multipath). Let's say ≈ 0.0003J. N0 energy becomes 0.5 - 0.0003 = 0.4997J.
    - e_rx_join(N4) = calculate_receive_energy(200) = 10e-6 J. N4 energy becomes 0.5 - 10e-6 = 0.49999J.
    - N0 added to Clusters[N4.id]. current_cluster_sizes[N4.id] becomes 1.
  - This process repeats for N1, N2, N3, N5, N6, N7, N9.
    - N1 joins N4. N4 members: [N0, N1]. Size: 2.

        - N2 joins N8. N8 members: [N2]. Size: 1.
        - N3 joins N4. N4 members: [N0, N1, N3]. Size: 3.
        - N5 joins N8. N8 members: [N2, N5]. Size: 2.
        - N6 joins N8. N8 members: [N2, N5, N6]. Size: 3.
        - N7 joins N4. N4 members: [N0, N1, N3, N7]. Size: 4.
        - N9 joins N4. N4 members: [N0, N1, N3, N7, N9]. Size: 5.
    - Each join involves energy deduction for both member and CH.
- **Output:** heact_clusters = {N4.id: [N0,N1,N3,N7,N9], N8.id: [N2,N5,N6]}, setup_energy_form (map of energy deducted for joins). ***(Illustration 2)***

4. Call to build_inter_cluster_tree_heact_further(or similar in run_simulation_protocol):

- Skipped because is_initial_phase is TRUE.

5. Call to simulate_heact_steady_state(nodes, heact_cluster_heads, heact_clusters, use_tree_phase=False):

- **Input:** nodes, heact_cluster_heads = [N4, N8], heact_clusters from above, use_tree_phase = False.
- **Internal Steps:**
    - ch_packets_aggregated = {N4.id: 0, N8.id: 0}.
    - **Phase 1 (Members to CH):**
        - For N4's cluster: N0, N1, N3, N7, N9 each transmit 4000 bits to N4.
            - Example N0->N4: d≈94m. E_Tx(N0) = calculate_transmit_energy(4000, 94). N0 energy decreases.
            - N4 receives 5 packets. E_Rx_total(N4) = 5 * calculate_receive_energy(4000). N4 energy decreases.
            - ch_packets_aggregated[N4.id] becomes 5.
        - For N8's cluster: N2, N5, N6 each transmit 4000 bits to N8. Similar energy deductions.
            - ch_packets_aggregated[N8.id] becomes 3.
    - **Phase 2 (CH Aggregation):**
        - N4: num_packets_from_members = 5. e_da = calculate_aggregate_energy(5*4000). N4 energy decreases.
        - N4: ch_packets_aggregated[N4.id] becomes 5+1=6 (adds self packet).

        - N8: num_packets_from_members = 3. e_da = calculate_aggregate_energy(3*4000). N8 energy decreases.
        - N8: ch_packets_aggregated[N8.id] becomes 3+1=4.
    - **Phase 3 (CH Transmission - Direct because use_tree_phase is False):**
        - N4: total_packets_to_send = 6. bits = 6*4000. d = N4.dist_to_bs ≈ 60.8m. E_Tx(N4) = calculate_transmit_energy(bits, 60.8). N4 energy decreases. packets_to_bs += 6.
        - N8: total_packets_to_send = 4. bits = 4*4000. d = N8.dist_to_bs ≈ 28.3m. E_Tx(N8) = calculate_transmit_energy(bits, 28.3). N8 energy decreases. packets_to_bs += 4.
- **Output:** steady_energy (map of energy consumed), round_packets = 10. ***(Illustration 3)***

Dry Run - Round 4 (Tree Phase)

1. Call to run_simulation_protocol (HEACT block):

- round_num = 4
- Condition (round_num - 1) % HEACT_RECLUSTER_INTERVAL == 0 is TRUE (3 % 3 == 0).
- heact_reconfig_count becomes 2.
- is_initial_phase = (2 <= 1) is FALSE.
- build_tree_this_round = False (will be set true later).

2. Call to select_heact_chs (...):

- **Input:** nodes (energies are now reduced, esp. for N4 and N8), p_ch=0.1, min_energy_threshold=0.1J, dist_factor=1.1.
- **Internal Steps:** Similar to Round 1, but probabilities for N4 and N8 will be much lower due to their depleted energy. Other nodes like N0, N1, N3, N5, N6, N7, N9 will have higher relative energy and thus higher chances.
- **Hypothetical Outcome:** Let's say N1 and N6 are selected as CHs (as in previous high-level dry run).
- heact_cluster_heads = [N1, N6].

3. Call to form_heact_clusters(nodes, heact_cluster_heads, HEACT_MAX_CLUSTER_SIZE):

- **Input:** nodes, heact_cluster_heads = [N1, N6], max_cluster_size = 15.
- **Internal Steps:** Non-CH nodes join N1 or N6.
- **Hypothetical Outcome:** heact_clusters = {N1.id: [N0,N3,N4,N7,N9], N6.id: [N2,N5,N8]}. Energy deducted for joins.

4. Call to build_inter_cluster_tree_heact_further(alive_chs_for_tree, bs_pos, HEACT_MIN_ENERGY_FOR_CH_RELAY, HEACT_TREE_ENERGY_PENALTY_EXPONENT):
    - is_initial_phase is FALSE.
    - **Input:** alive_chs_for_tree = [N1, N6] (assuming they survived cluster formation), bs_pos=(50,150), ch_relay_energy_threshold = 0.2J, energy_penalty_exponent = 1.5.
    - **Internal Steps:**
        - Assume N1.energy > 0.2J and N6.energy > 0.2J (they are eligible relays).
        - sorted_chs = [N6, N1] (N6 is closer to BS: N6.dist_to_bs ≈ 31.6m, N1.dist_to_bs ≈ 80.6m).
        - primary_root = N6.
        - N6.parent_ch_id = "BS". N6.path_cost_sq_ch = N6.dist_to_bs^2 ≈ (31.6)^2 ≈ 998.56. N6.is_relay_ch = True. visited_ch_ids = {N6.id}. root_chs = [N6].
        - current_tree_potential_parents = {N6.id: N6}.
        - unvisited_chs = [N1].
        - **Iteration to add N1:**
            - q_ch = N1. i_ch = N6.
            - dist_qi(N1,N6) = calculate_distance((30,80), (20,140)) ≈ 60.83m.
            - EnergyRatio(N6) = max(0.01, N6.energy / 0.5). Assume N6.energy is 0.45J (after some CH selection/formation cost). EnergyRatio(N6) = 0.45/0.5 = 0.9.
            - Penalty(N6) = (1.0 / 0.9) ** 1.5 ≈ 1.111 ** 1.5 ≈ 1.17.
            - current_cost = ((60.83^2) + 998.56) * 1.17 ≈ (3700.29 + 998.56) * 1.17 ≈ 4698.85 * 1.17 ≈ 5497.6.
            - min_cost = 5497.6. best_next_ch = N1. best_parent_ch_id = N6.id.
        - N1 is added: N1.parent_ch_id = N6.id. N6.children_ch_ids.add(N1.id). N1.path_cost_sq_ch = 5497.6.
        - N1.is_relay_ch = (N1.energy >= 0.2J) (TRUE). current_tree_potential_parents adds N1.
        - visited_ch_ids.add(N1.id). unvisited_chs is now empty. Loop terminates.
    - build_tree_this_round = True.
    - **Output:** CHs N1 and N6 have their tree attributes updated. ***(Illustration 4)***

Call to simulate_heact_steady_state(nodes, heact_cluster_heads, heact_clusters, use_tree_phase=True):

- **Input:** nodes, heact_cluster_heads = [N1, N6], heact_clusters, use_tree_phase = True.
- **Internal Steps:**
  - ch_packets_aggregated = {N1.id: 0, N6.id: 0}.
  - **Phase 1 (Members to CH):**
    - N1's members ([N0,N3,N4,N7,N9]) transmit to N1. N1 energy decreases from 5 receptions. ch_packets_aggregated[N1.id] becomes 5.
    - N6's members ([N2,N5,N8]) transmit to N6. N6 energy decreases from 3 receptions. ch_packets_aggregated[N6.id] becomes 3.
  - **Phase 2 (CH Aggregation):**
    - N1 aggregates 5 packets. e_da deducted. ch_packets_aggregated[N1.id] becomes 5+1=6.
    - N6 aggregates 3 packets. e_da deducted. ch_packets_aggregated[N6.id] becomes 3+1=4.
  - **Phase 3 (CH Transmission - Tree because use_tree_phase is True):**
    - ch_traversal_order = [N1.id, N6.id] (N1 is leaf, N6 is root).
    - **Process N1:**
      - total_packets_to_send = 6. bits = 6*4000. parent_id = N6.id.
      - d(N1,N6) ≈ 60.83m. E_Tx(N1) = calculate_transmit_energy(bits, 60.83). N1 energy decreases.
      - N6 receives 6 packets (24000 bits) from N1. E_Rx(N6) = calculate_receive_energy(24000). N6 energy decreases.
      - ch_packets_aggregated[N6.id] becomes 4 (from members+self) + 6 (from N1) = 10.
    - **Process N6:**
      - total_packets_to_send = 10. bits = 10*4000. parent_id = "BS".
      - d(N6,BS) ≈ 31.6m. E_Tx(N6) = calculate_transmit_energy(bits, 31.6). N6 energy decreases significantly.
      - packets_to_bs += 10.
- **Output:** steady_energy, round_packets = 10.

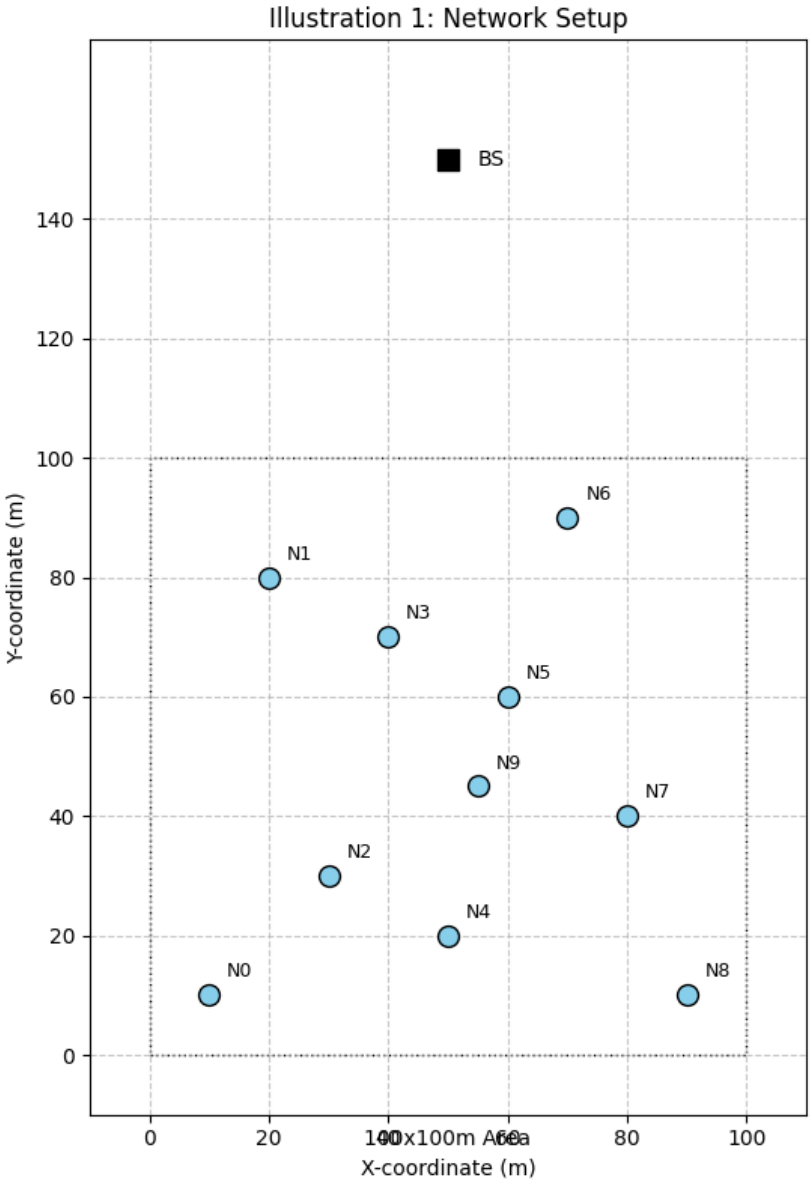
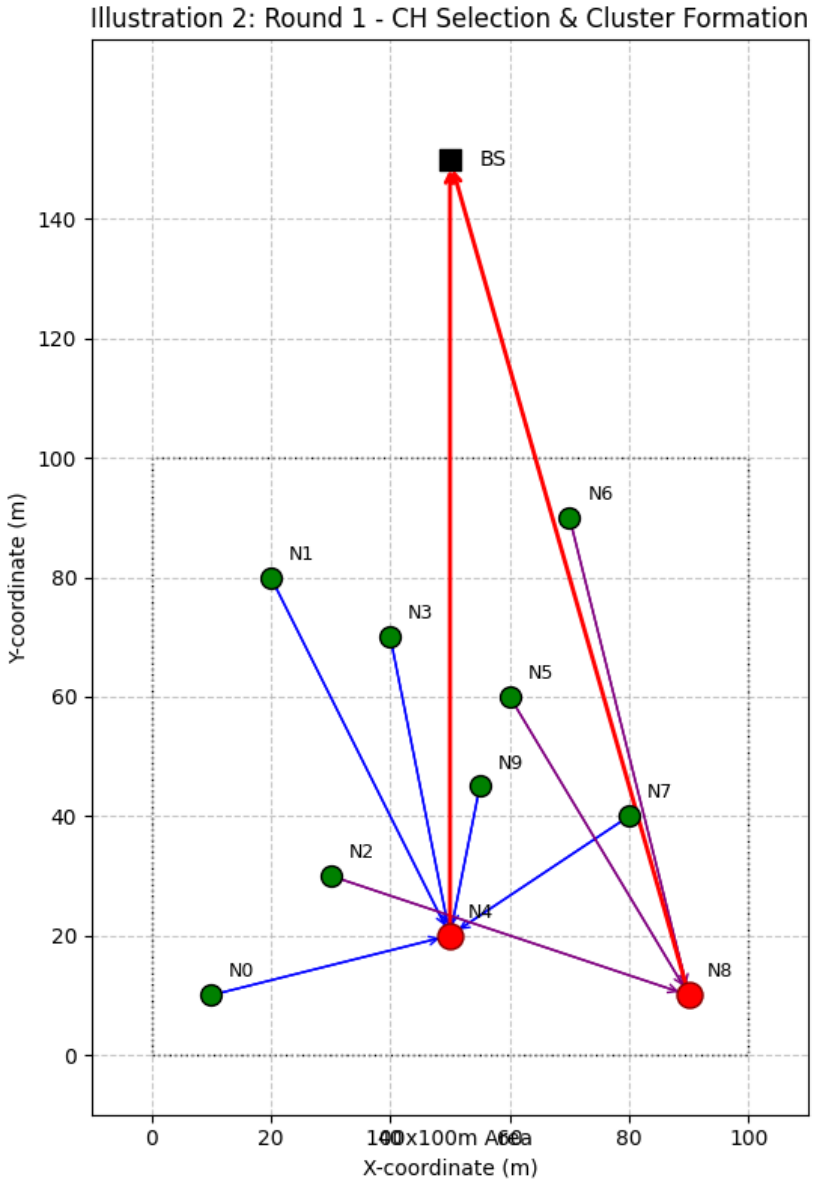
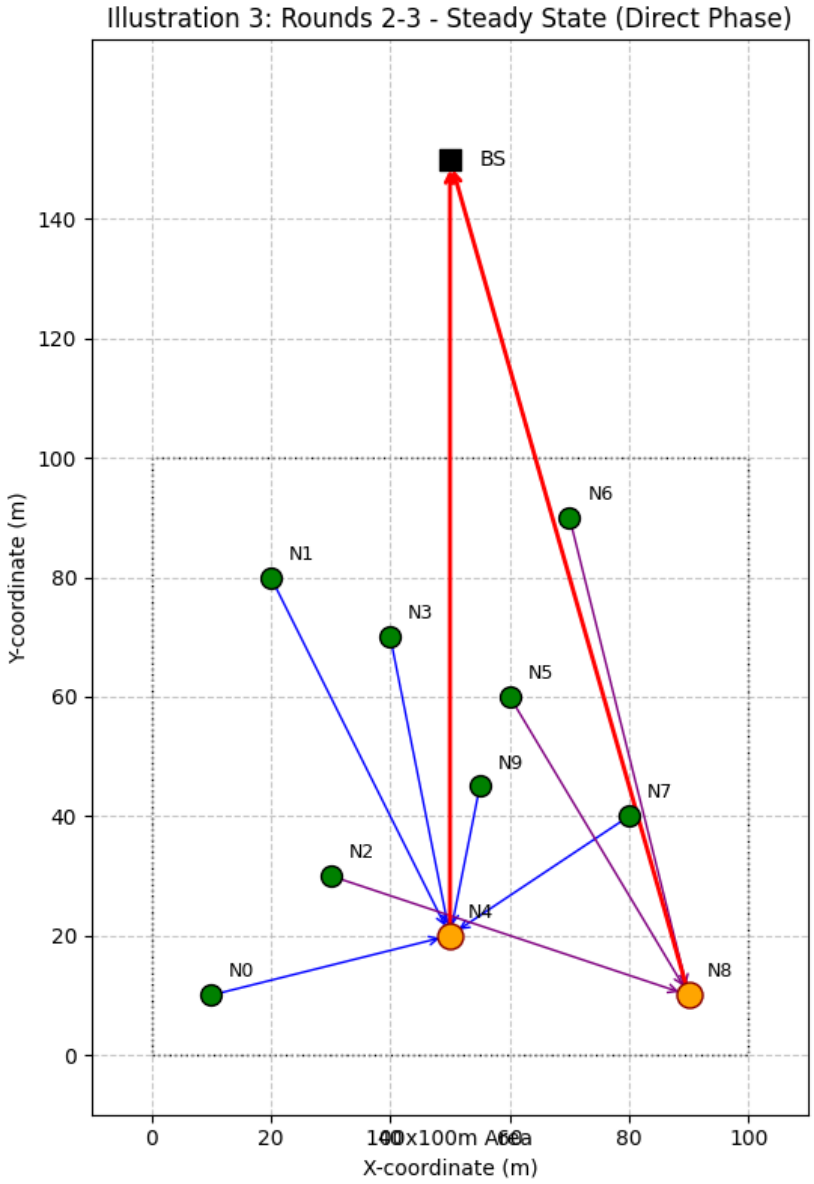


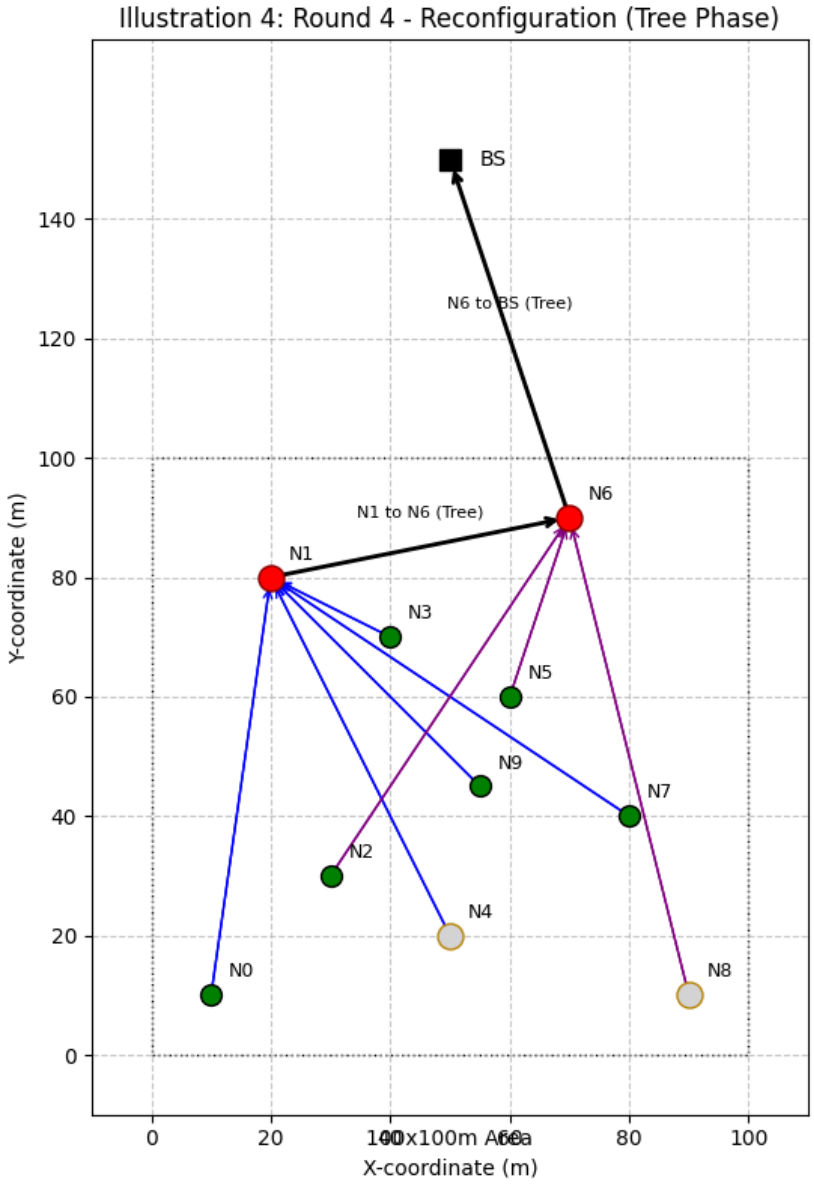
Illustration 1: Network Setup
BS
N0
N1
N2
N3
N4
N5
N6
N7
N8
N9
Y-coordinate (m)
X-coordinate (m)
0
20
40
60
80
100
120
140
100x100m Area



Round 1: Initial Direct Phase. CHs: N4, N8. Members join nearest CH.
CHs transmit aggregated data directly to BS.



Rounds 2-3: Steady State (Direct Phase). CHs N4, N8 continue with direct BS transmission. Energy depletes.



Round 4: Reconfiguration (Tree Phase). New CHs: N1, N6.
Inter-CH Tree: N1->N6->BS.