Chapter 6

Networking Resource Allocation for Group Communication

The previous chapters proposed approaches for allocating the edge computing resources of the multi-tenant edge servers for processing the offloaded tasks. In this chapter, we propose a method for allocating edge networking resources for group communication.

6.1 Introduction

In IoT, multiple sensor nodes may collaborate at the sensing layer to collect data from their environment [29]. For example, as shown in Fig.6.1, temperature sensing nodes (blue round drum shape nodes) can work collaboratively to satisfy the environmental requirements. In this process, any temperature sensing node may need to send its sensing data through different nodes to other temperature sensing nodes. Thus, IoT nodes collectively acquire the data and transmit it to the data management layer for further processing [29]. The intelligent service layer then takes action based on processed data and provides the required information to various destinations. For an example of a fire scenario (Fig.6.1), the edge server processes the data transmitted by temperature sensing nodes to detect fire. It acts as a source node for a rescue operation, disseminating

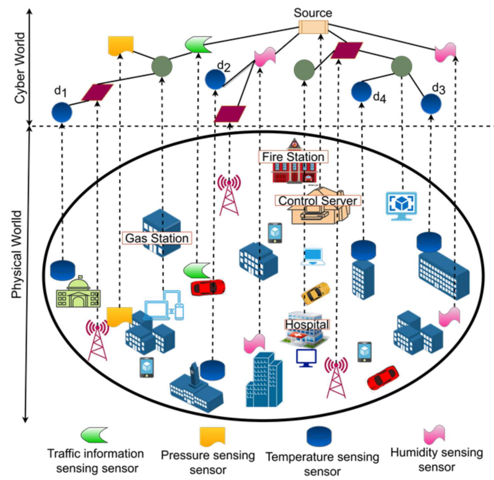


Figure 6.1: An example of a smart city.

the information needed to various destinations (fire station, gas station, police station, etc.) to provide services —The other devices (sensors, stations, etc.) help forward the messages for the rescue operation. In such cases where nodes are grouped, multicast is more efficient than broadcast and unicast at network and data sensing layers [30].

In multicast routing, a source node provides the feed to multiple destination nodes simultaneously; all the destination nodes establish connections to the source node like many-to-one network communication [141]. Fig.6.2 represents the two scenarios of multicast: (a) a source node s multicast the single-data unit to destination nodes (d_1, d_2, d_3, d_4) in the homogeneous networks, and (b) a source node s multicast the multiple-data units (D_1, D_2) to destination nodes (d_1, d_2, d_3, d_4) in the heterogeneous networks [142]. In heterogeneous networks, each network uses a different protocol for data transmission (for example, one uses a wireless sensor network protocol for data transmission, and the other uses TCP/IP protocol). Fig.6.2 shows that the intermediate nodes create the paths dynamically from source to destination in wireless networks as it increases the efficiency of the whole network [143]. These paths can be selected based on various Quality of Service (QoS) requirements of the underlying application, such as throughput, delay, energy consumption, jitter, etc.

Generally, directed graphs represent the underlying network where edges represent the link among the nodes (IoT devices, routers, data processing units, etc.), and each

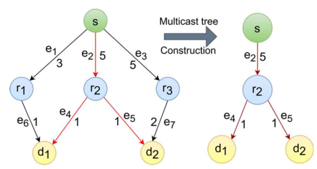
(a) (b)

Figure 6.2: Figure shows (a) single-data unit multicast, and (b) multiple-data units multicast.

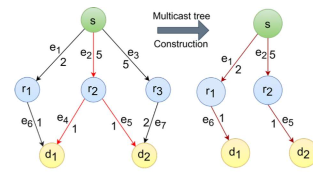
edge has a nonnegative cost (depends on network metrics such as throughput, energy consumption, delay, etc.), as shown in Fig.6.3. Each destination node bears the cost of the edges in its path to receive the feed from the source node. If the path to more than one destination node in a multicast group includes an edge, the user destination nodes may share its cost. In this case, the cost of a path for any destination node also depends on other destination nodes' paths.

A better QoS with effective utilization of resources can be provided by constructing the lowest cost multicast tree for group communication. We are considering multiple objectives for building a cost-effective multicast tree, e.g., high throughput, low delay, low energy consumption, etc. These multiple objectives can be efficiently combined in the form of edge costs. Therefore, it is required to construct the lowest cost tree based on the cost of edges, which includes the source node, all destination nodes, and some additional intermediate nodes (required to build the minimum cost tree between source to destinations). This type of lowest cost tree that includes a specific subset of nodes with some additional nodes is called Minimum Steiner Tree (MST) [31]. Thus, one of the approaches to achieve the least cost multicast tree based on the edge costs is to form the MST [32–34]. Hence, in this chapter, we propose a distributed approach to construct the MST for group communication.

This chapter proposes a game-theoretic approach that solves the problem of forming the cost-effective multicast tree for group communication. We use the different physical properties of the network, e.g., bandwidth, physical distance between nodes,



(a)



(b)

Figure 6.3: The same network with different costs of edges. (a) Case I. (b) Case II.

transmission range, etc., to calculate the cost of an edge. The cost assigned to any edge is inversely proportional to the edge's performance in terms of QoS, which means that a lower-cost edge provides better QoS. We then optimize the cost of the multicast tree by proposing a cost-sharing scheme for network edges. The scheme constructs an optimal multicast tree by increasing the resource-sharing among the destination nodes in the network. The main contributions of this chapter are as follows:

- Constructing a cost-effective multicast tree for group communication is modeled as a Path Selection Game (PSGame). In this game, each destination node is considered as a player.
- The cost optimization problem of the multicast tree is formulated in terms of the individual cost functions of the destinations nodes. The cost function of each destination node defines the path cost between the source to the destination. Each destination node aims to find the minimum cost path.
- A cost-sharing scheme is proposed to divide the cost of a network edge among the edge's users. The destination nodes collectively reach the optimal PNE under this cost-sharing scheme.
- A decentralized Path Selection Algorithm (PSA) is proposed to achieve the PNE in PSGame as a stable solution.
- PSGame is proved as a potential game that admits at least one PNE using the PSA under the proposed cost-sharing scheme.
- The performance of the proposed algorithm is analyzed both theoretically and numerically.

The rest of the chapter is constituted as follows: Section 6.2 motivates this work through an example. Section 6.3 gives the system model. Section 6.4 presents the cost-sharing scheme and the PSA. Section 6.5 shows the existence of Nash equilibrium. Section 6.6 and 6.8 assess the PSA theoretically and numerically.

6.2 Motivating Example

To understand the optimization problem for path selection in multicast, we consider two scenarios as given in Fig. 6.3a and Fig. 6.3b. In both scenarios, a source node s feeds the destination nodes d_1 and d_2. In Fig.6.3a, if both d_1 and d_2 get feed through the middle edge e_2, then$\{e_2, e_4\}$ and $\{e_2, e_5\}$ will be the paths from s to d_1 and d_2, respectively. The subgraph formed by these paths is a MST $\{e_2, e_4, e_5\}$, and all destination nodes' total cost of receiving feeds through this tree is 7. If fair-division cost-sharing scheme derived from the Shapely value [1] [52] is used to split the edge cost among its users, each d_1 and d_2 incurs the cost 3.5 for paths $\{e_2, e_4\}$ and $\{e_2, e_5\}$, respectively. The paths ($\{e_2, e_4\}$, $\{e_2, e_5\}$) for d_1 and d_2 are also PNE because no destination node can do better by changing its path. Therefore, global optimum and PNE are identical for the network given in Fig. 6.3a.

In Fig.6.3b, destination node d_1 has a path $\{e_1, e_6\}$ with cost 3 to get the information from source node s. Destination node d_1 would prefer to receive information via path $\{e_1, e_6\}$ because the cost of path $\{e_1, e_6\}$ (3) is less than the cost of path $\{e_2, e_4\}$ (3.5 as explained earlier). Therefore, destination node d_1 will not share edge e_2 with destination node d_2. Thus, destination nodes d_1 and d_2 will receive information through paths $\{e_1, e_6\}$ and $\{e_2, e_5\}$, respectively. The paths ($\{e_1, , e_6\}$, $\{e_2, e_5\}$) form a PNE with edges $\{e_1, e_2, e_5, e_6\}$ because no destination node can do better by changing its path unilaterally. The total cost of all destination nodes at PNE is 9, while the total cost of the global optimum solution is 7. It observes that the unique PNE differs from the

[1] The Shapley value is a solution concept applied in game theory that involves fairly dividing both gains and costs to several players working in a coalition.

global optimum. In such cases, it is interesting to examine whether the total cost at PNE can be further reduced and what cost-sharing scheme can be used for this. In this chapter, we propose a cost-sharing scheme that motivates the destination nodes to more edge-sharing.

6.3 System Model

To formulate the multicast routing problem, we model the underlying network as a graph $G(V, E)$, where V and E are node-set and edge-set, respectively, as shown in Fig. 6.3. The node-set can be partitioned into three different categories: 1) destination node-set U (sensor nodes, security management devices, smart phone, etc.), 2) intermediate or routing node-set R, and 3) source node-set X (rendezvous point, control server, sensor node, base station, etc.). We assume that IoT nodes are equipped with limited power batteries and transmit data to other nodes via wireless means. An edge represents a link between two nodes, specified by e with a nonnegative cost $\varsigma(e)$.

The edge cost between two nodes can be modeled differently according to the application's requirements as a function of QoS parameters. In this chapter, the cost of an edge e is modeled inversely proportional to the throughput while proportional to the delay and energy consumption. This means that a low-cost edge has overall high throughput, low delay, and lower energy consumption. Thus, edge cost $\varsigma(e)$ between any two nodes, u and v, is defined as follows:

$$\varsigma(e) = \kappa_1 \tau(e) + \kappa_2 \delta(e) + \kappa_3 \beta(e) \tag{6.1}$$

Where cost component $\tau(e)$ depends on the bandwidth, $\delta(e)$ depends on the distance between nodes, and $\beta(e)$ depends on the energy consumption by nodes u and v to transmit and receive the data, respectively. $(\kappa_1, \kappa_2, \kappa_3)$ are weights, such as $\kappa_1+\kappa_2+\kappa_3 = 1$. According to the application's requirements, these weights can be adjusted for tuning the tradeoffs among throughput, delay, and energy consumption.

$\tau(e)$ for an edge e can be defined as follows:

$$\tau(e) = \frac{R_\tau}{B_e} \tag{6.2}$$

Where B_e is the bandwidth of edge e between nodes u and v, and R_τ is the reference bandwidth used to normalize the bandwidth. As the bandwidth between two nodes increases, the throughput will also increase, and the transmission delay will reduce. Since $\tau(e) \propto \frac{1}{B_e}$, an edge with a lower $\tau(e)$ value has a high throughput and a low transmission delay.

$\delta(e)$ for an edge e is defined as follows:

$$\delta(e) = \frac{L_e}{\vartheta \times M_\delta} \tag{6.3}$$

Where L_e is the distance between two nodes, ϑ is the signal speed, and M_δ is the propagation delay between any two nodes with minimum distance in the network. Delay M_δ can be calculated as $\min_{f \in E} \ (\frac{L_f}{\vartheta})$ and used to normalize the delay. If distance L_e between two nodes increases, the propagation delay will decrease. Since, $\delta(e) \propto L_e$, an edge with a low $\delta(e)$ value has a lower propagation delay and vice versa.

$\beta(e)$ for an edge e can be calculated as follows:

$$\beta(e) = \frac{\xi(e)}{M_\xi} \tag{6.4}$$

Where $\xi(e)$ is the energy required to transmit a packet via edge e, and M_ξ is the required energy for data transmission between any two nodes with minimum distance in the network such as $\min_{f \in E} \ (\xi(f))$. Since $\beta(e) \propto \xi(e)$, an edge with a lower $\beta(e)$ value has lower energy consumption and vice versa.

The energy consumption $\xi(e)$ in transmitting b bit data via an edge e will be as follows:

$$\xi(e) = \xi_{trans}(L_e, b) + \xi_{recv}(b) \tag{6.5}$$

Where ξ_{trans} and ξ_{recv} are energy consumption by node u for transmitting data and by node v for receiving, respectively. If L_{th} is a given threshold and $L_e < L_{th}$, we use the free space model (L_e^2) otherwise multi-path model (L_e^4) is used [53]. Thus the energy consumption at node u for transmitting data over an edge e is,

$$\xi_{trans}(L_e, b) = \begin{cases} b\xi_{td} + b\xi_{fa}L_e^2 & : L_e \leq L_{th} \\ b\xi_{td} + b\xi_{ae}L_e^4 & : L_e > L_{th} \end{cases} \tag{6.6}$$

Where $b\xi_{td}$ specify the energy consumed by transmitter electronic circuit for b bits data. $b\xi_{fa}L_e^2$ and $b\xi_{ae}L_e^4$ denote the transmitting amplifier energy for b bits data depends on distance L_e. The energy consumption by node v for receiving b bits:

$$\xi_{recv}(b) = b\xi_{td} \tag{6.7}$$

For simplicity, we assume that source node-set X contains a single source node, which multicasts a single data unit and multiple-data units into homogeneous and heterogeneous networks, as shown in Fig.6.2 and Section 6.1. There may be more than one source node in X; the same multicast tree construction algorithm can be easily extended for multiple sources.

6.3.1 Path Selection Game

The cost of any edge is apportioned to its user nodes, as specified in the example above (Section 6.2). Therefore, the path cost incurred by the destination node depends not only on its path but also on other destination nodes' paths. If a destination node changes its path, then it may also affect the cost of other destination nodes. Thus, there is a strategic interaction among the destination nodes where the cost incurred by a destination node depends on its path and other destination nodes' paths. Hence, the path selection process from source to destination nodes can be modeled as a game where each destination node behaves as a player. The PSGame is defined as follows:

Destination Nodes $U = \{1, 2, \cdots, n\}$ is a n destination node-set where each destination node i act as a player.

Path Set Each destination node i has at least one simple path, denoted by p_i, to get the feed from the source node. A finite set P_i represents all the paths from the source node to any destination node i.

Path Selection Profile A tuple of simultaneous paths constructed by all destination nodes to get data from the source node is called a path selection profile specified by $p = (p_1, p_2, \cdots, p_n)$, where every $p_i \in P_i$ is a path constructed by a destination node i. To deal with any specific destination node i, the path selection profile can be written as $p = (p_i, p_{-i})$, where p_i is the path of i and $p_{-i} = \{p_1, p_2, \cdots p_{i-1}, p_{i+1}, \cdots, p_n\}$ is a path list of other destination nodes except i.

Cost Function of Destination nodes Given network graph $G(V, E)$ and $p = (p_i, p_{-i})$, function $c_i(p)$ defines the cost incurred by a destination node i to get data via path p_i. Each destination node i's preference is to select a path $p_i \in P_i$ that minimizes the cost $c_i(p)$ in response to other destination nodes' paths p_{-i}. Suppose for each edge $e \in E$; a list S_e contains the destination nodes getting the feed from the source via e. The cost $\varsigma(e)$ of an edge e is divided among its user destination nodes in S_e, such that $\varsigma(e) = \sum_{i \in S_e} c_i^e$, where c_i^e is the cost paid by the destination node i for using e. Thus, the cost incurred by any destination node i for a path p_i at path selection profile p is calculated as $c_i(p) = \sum_{e \in p_i} c_i^e$. We assume that the information related to any edge e and destination nodes S_e (getting feed through edge e) are stored at nodes at both ends of edge e.

6.3.2 Optimization Model

In the PSGame, the destination nodes make path selection decisions, following the game's rules, to accomplish their objectives. At a given path selection profile, each destination node aims to switch on an alternative path that minimizes its cost, and the PSGame transits from one state to another in the form of a path selection profile. Given destination nodes $U = \{1, 2, \cdots, n\}$, network graph $G(V, E)$, and other destination nodes path list p_{-i}, a destination node i constructs a suitable path p_i to minimize cost $c_i(p)$:

$$\min_{p_i \in P_i} c_i(p_i, p_{-i}) \tag{6.8}$$

Let the multicast tree formed by paths of all the destination nodes is defined as $N(V', E') = \cup_{i=1}^{n} p_i$ at a path selection profile p, then the overall cost of destination nodes for a multicast tree $N(V', E')$ at p is,

$$\zeta(p) = \sum_{e \in E'} \varsigma(e) \tag{6.9}$$

This chapter aims to find a path selection profile at which the total cost of all destination nodes is a stable optimum. This stable optimum solution is a Pure Nash Equilibrium (PNE) that is achieved by optimizing each destination node's cost function.

Definition 6.1 (Pure Nash Equilibrium) *In PSGame, an path selection profile $p^* = (p_i^*, p_{-i}^*)$ is a PNE if no destination node can further reduce his cost by unilaterally changing his path, i.e.,*

$$c_i(p_i^*, p_{-i}^*) \leq c_i(p_i, p_{-i}^*), \forall p_i \in P_i, \forall i \in U \tag{6.10}$$

Whether the PSGame admits at least one PNE using the proposed path selection procedure is of vital importance to this work. Section 6.5 examines the existence of a

PNE in the PSGame.

6.4 Multicast Tree Construction

This section presents the cost-sharing scheme and a Path Selection Algorithm (PSA) with an illustration by which the system converges to PNE in PSGame. In PSGame, every destination node selects its path that reduces its cost in response to others. Due to which all the destination nodes gradually move towards a PNE while continuously searching for their best path.

Definition 6.2 (Best Path) *Given the path selection profile p_{-i} of all the receiver nodes except i, the best path of node i is the path p'_i that minimizes his cost in response of other nodes, defining as:*

$$BP_i(p_{-i}) = \{p'_i \in P_i : c_i(p'_i, p_{-i}) \leq c_i(p_i, p_{-i}), \forall p_i \in P_i\} \tag{6.11}$$

6.4.1 Cost-sharing Scheme

The proposed cost-sharing scheme divides the edge cost among the edge user destination nodes in an incentive-based unequal fashion. For example, as discussed above (Fig.6.3b) in Section 6.2, destination nodes d_1 and d_2 do not share edge e_2 because sharing does not minimize the cost of d_1. As a result, the system's overall cost at Nash equilibrium is greater than the central optimum. Our proposed cost-sharing scheme gives an incentive that motivates node d_1 to share the edge e_2 with node d_2. This scheme considers the current path cost of each destination nodes in a multicast group as their weight. These weights are used to divide the cost of future edges, which motivates the destinations nodes of less path cost to switch on such paths where more edges are shared with others than before. More elaborately, if $p = (p_1, p_2, \cdots, p_n)$ is the existing path selection profile at any moment, then the destination nodes' respective path costs in $(c_1(p), c_2(p), \cdots, c_n(p))$ work as their weights at that game instance.

Initially, when nodes start constructing routes to receive feed from the source or new nodes are added to the network, they do not have current paths in either case. The default weight assigned to each destination node is assumed one.

In the proposed path construction approach for multicast, the destination nodes start with an arbitrary path selection profile and keep changing the paths to minimize their cost until the system reaches the PNE. Initially, each destination node has weight one, so the edge's cost $\varsigma(e)$ is divided equally among the users of e such that the cost paid by each $i \in S_e$ for e is,

$$c_i^e = \varsigma(e)/|S_e| \tag{6.12}$$

The path cost for each destination node will be,

$$c_i(p) = \sum_{e \in p_i | p_i \in p} \varsigma(e)/|S_e| \tag{6.13}$$

After the initialization, suppose a destination node changes its path in order to minimize his cost and the path selection profiles changes from p to p'. While changing the path, destination nodes leave some edges λ_{ex} and include some new edges λ_{in} to its path. Each user destination node i of each edge $e \in \lambda_{ex} \cup \lambda_{in}$ share the edge's cost $\varsigma(e)$ as,

$$c_i^e = \varsigma(e) \cdot \frac{c_i(p)}{\sum_{k \in S_e} c_k(p)}, \text{ for every } i \in S_e \tag{6.14}$$

The updated cost of a destination node i for path $p_i \in p'$ is,

$$c_i(p') = \sum_{e \in p_i | p_i \in p'} \varsigma(e) \cdot \frac{c_i(p)}{\sum_{k \in S_e} c_k(p)} \tag{6.15}$$

6.4.2 Path Selection Algorithm (PSA)

Given destination node-set $U = \{1, 2, \cdots, n\}$ and network graph $G(V, E)$, the PSGame employs an iterative process to achieve the PNE as given in Algorithm 6.1. In this process, all the destination nodes initially select arbitrary paths (Line 1). The nodes at

Algorithm 6.1: Path Selection Algorithm

Input: $U, G(V, E)$
Output: Path selection profile p

1 **Initialize** $p \leftarrow (p_1, p_2, p_3, \cdots p_n)$ to be an arbitrary path selection profile
2 nodes at both ends of every edge $e \in E$ keep a list S_e of the destination nodes getting feed via e in a multicast group
3 each destination node $i \in U$ calculates its path cost by Eq.6.13 and communicates to hops in its path
4 **repeat**
5   **for** *each destination node* $i \in U$ **do**
6   **for** *each path* $p_i \in P_i$ **do**
7   calculate the cost of path p_i at path selection profile $p(t)$ using Eq.6.15
8   **end**
9   find the path p_i' that incurs the lowest path cost accros P_i
10   **if** $p_i(t) \neq p_i'$ **then**
11   send request to participate in path update from $p_i(t)$ to p_i'
12   **if** *request is selected* **then**
13   update S_e for all edges in $\lambda_{ex} \cup \lambda_{in}$ as Eq.6.16
14   i changes its path with p_i'
15   every destination node $j \in U$ calculate its path cost by Eq.6.15 and comunicate to hops in its path
16   **end**
17   **end**
18   **end**
19 **until** *no more path update required*

both ends of each edge $e \in E$ maintain S_e containing the list of destination nodes of the multicast group who use edge e to get feed from the source node (Line 2). Then each destination node calculates the cost of its path as given in Eq.6.13 and communicates the path cost to all hops in its path (Line 3). After the initialization, an iterative process starts to find the PNE, and the computation is done in parallel by destination nodes in each iteration of the path selection process (Line 4-19). Every iteration is denoted by $t(t = 1, 2, 3....)$, and path selection profile p at any iteration t is specified by $p(t)$.

In every iteration of the process, each destination node i calculates the cost of every path $p_i \in P_i$ by using Eq.6.15 (Line 6-8). Next, each destination node i finds a path p_i' at the path selection profile $p(t)$, which incurs a minimum cost (Line 9). If destination node

i's current path $p_i(t)$ is not a p'_i, it requests path update participation (Line 10-11). In each iteration, only one destination node's request is selected in a decentralized manner which requires message synchronization [144]. If any destination node i's request is selected, it updates his path with p'_i. During path update, the destination node i leaves the edges $\lambda_{ex} = p_i(t) \setminus p'_i$ and adds edges $\lambda_{in} = p'_i \setminus p_i(t)$ to its path. For every edge $e \in \lambda_{ex} \cup \lambda_{in}$, the destination node-set S_e is updated as,

$$S_e = \begin{cases} S_e \setminus \{i\} : \forall e \in \lambda_{ex} \\ S_e \cup \{i\} : \forall e \in \lambda_{in} \end{cases} \tag{6.16}$$

After updating the S_e (Line 13), destination node i sets its path with p'_i (Line 14). Finally, all the destination nodes calculate their selected paths cost according to Eq.6.15 and communicate them to all the hops in their path (Line 15). In this way, the PSA's iterations repeat for destination nodes until they do not want to change their path. Eventually, each destination node's path makes a final path selection profile with the PNE (stable solution).

6.5 Game Property

In this section, we investigate weather the PSGame admits a PNE under the proposed PSA. The key is to prove that PSGame is a potential game defined as:

Definition 6.3 (Potential Game) *Given a game, if there is a potential function* $\Phi(p)$ *such that,*

$$c_i(p'_i, p_{-i}) \leq c_i(p_i, p_{-i}) \Rightarrow \Phi(p'_i, p_{-i}) \leq \Phi(p_i, p_{-i}) \tag{6.17}$$
$$\textit{for any } i \in U, p_i, p'_i \in P_i \textit{ and } p_{-i} \in \prod_{l \neq i} P_l$$

In this chapter, a function Φ for PSGame is defined as,

Algorithm 6.2: Potential Function Value

Input: $U, G(V, E)$
Output: Potential Function value $\Phi(p)$

$\Phi(p) \leftarrow 0$
for *each edge* $e \in E$ **do**
  $Q \leftarrow S_e$, $F(S_e) \leftarrow 0$, $B \leftarrow \emptyset$
  while $Q \neq \emptyset$ **do**
    extract l from Q s.t. $\forall j \in Q$: $c_l(p) \geq c_j(p)$
    $B \leftarrow B \cup l$
    select i from B s.t. $\forall k \in B$: $c_k(p) \geq c_i(p)$
    $F(S_e) \leftarrow F(S_e) + \frac{c_i(p)}{\sum_{k \in B} c_k(p)}$
  end
  $\Phi(p) \leftarrow \Phi(p) + \varsigma(e) \cdot F(S_e)$
end

$$\Phi(p) = \sum_{e \in E} \varsigma(e) \cdot F(S_e) \tag{6.18}$$

where $F(S_e)$ for an edge e is calculated as,

$$F(S_e) = \sum_{i \in S_e} \frac{c_i(p)}{\sum_{k \in S_e, c_k(p) \geq c_i(p)} c_k(p)}. \tag{6.19}$$

The potential function $\Phi(p)$ at path selection profile p in PSGame can be determined as in Algorithm 6.2.

Theorem 6.1 *The PSGame is a potential game with potential function $\Phi(p)$.*

Proof: We assume that current path selection profile is p' which evolved from the path selection profile p under the PSA. Let a destination node i changes its path from p'_i to p''_i as the best response and consequently path selection profile changes from p' to p''. The destination node i see it as an improvement which implies that the path change will lead to decrement as,

$$c_i(p'') < c_i(p') \tag{6.20}$$

While changing the path, destination i excludes edges $\lambda_{ex} = p'_i - p''_i$ and includes edges $\lambda_{in} = p''_i - p'_i$. Hence, overall effect on destination node will be only by edges in λ_{ex} and λ_{in}. From Eq. 6.14 and Eq.6.20,

$$\sum_{f \in \lambda_{in}} \varsigma(f) \cdot \frac{c_i(p')}{\sum_{k \in S_f} c_k(p')} < \sum_{g \in \lambda_{ex}} \varsigma(g) \cdot \frac{c_i(p)}{\sum_{k \in S_g} c_k(p)} \tag{6.21}$$

Consequently, the change in the value of function Φ

$$\begin{aligned} \Delta\Phi &= \text{increment in potential} - \text{decrement in potential} \\ &= \Phi_{inc} - \Phi_{dec} \end{aligned} \tag{6.22}$$

The edges in λ_{in} added to the path by the destination node i increase the game's potential function $\Phi(.)$. Hence, the increase in potential from Eq.6.18 is $\Phi_{inc} = \sum_{f \in \lambda_{in}} c(f) \cdot F(S_f)$. By expending the term $F(S_f)$ from Eq.6.18,

$$\begin{aligned} \Phi_{inc} &= \sum_{f \in \lambda_{in}} \varsigma(f) \cdot \frac{c_i(p'')}{\sum_{k \in S_f, c_k(p'') \geq c_i(p'')} c_k(p'')} \\ &< \sum_{f \in \lambda_{ex}} \varsigma(f) \cdot \frac{c_i(p'')}{\sum_{k \in S_f} c_k(p'')} \end{aligned} \tag{6.23}$$

The edges in λ_{in} dropped by the destination node i decrease the game's potential function $\Phi(.)$. Hence, the decrease in potential from Eq.6.18 is $\Phi_{dec} = \sum_{g \in \lambda_{ex}} c(g) \cdot F(S_g)$. By expending the term $F(S_g)$ from Eq.6.18,

$$\begin{aligned}\Phi_{dec} &= \sum_{g \in \lambda_{ex}(i)} \varsigma(g) \cdot \frac{c_i(p')}{\sum_{k \in S_g, c_k(p') \geq c_i(p')} c_k(p')} \\ &< \sum_{g \in \lambda_{ex}} \varsigma(g) \cdot \frac{c_i(p')}{\sum_{k \in S_g} c_k(p')} \\ &\quad \text{from the Eq.6.20, } c_i(p'') < c_i(p'), \text{ so} \\ &> \sum_{f \in \lambda_{ex}} \varsigma(f) \cdot \frac{c_i(p'')}{\sum_{k \in S_f} c_k(p'')} \end{aligned} \tag{6.24}$$

Thus, the change in potential $\Delta\Phi$ defined in Eq.6.22, given the inequalities bounders in Eq.6.23 and Eq.6.24 will be < 0. Hence, $c_i(p''_i) < c_i(p'_i) \Rightarrow \Phi(p'') < \Phi(p')$. Therefore, PSGame is a potential game with potential function $\Phi(p)$. □

Theorem 6.2 *The PSGame admits at least one Pure Nash Equilibrium (PNE).*

Proof: In the PSGame, each destination node has a finite number of shortest paths to get feed from the source node; therefore, PSGame has a finite number of path selection profiles. As proved in Theorem 6.1, the potential function monotonically decreases with every PSA iteration; so, there is no cycle possible during the execution of the PSA. As a result of both these reasons, the path selection process under the PSA eventually stops. If the process stops, it means that none of the destination nodes have any beneficial paths to deviate from its current path unilaterally, which is the state of the PNE in the PSGame. □

6.5.1 Convergence Analysis

The PSGame admits at least one PNE under the PSA. Let Q are the total number of iteration, $\Phi_{max} := \max(\Phi(p))$, $\Phi_{min} := \min(\Phi(p))$, average cost of edges $\varsigma_{avg} := \text{avg}(\varsigma(e))$, maximum number of paths from source to any destination node $r_{max} := \max(|P_i|)$, and average path length between source to destination nodes l_{avg}. The maximum value of potential function will be when destination nodes have no shared edges in their paths, Φ_{max} will be,

$$\Phi_{max} = \sum_{e \in p_i | p_i \in p} \varsigma(e) \cdot F(S_e) \tag{6.25}$$

In this case, $|S_e| = 1$ for each edge that implies $F(S_e) = 1$. Thus

$$\Phi_{max} = \sum_{e \in p_i | p_i \in p} \varsigma(e) = \sum_{e \in p_i | p_i \in p} \varsigma_{avg} = n.l_{avg}.\varsigma_{avg} \tag{6.26}$$

The minimum possible value of $\Phi(p)$ will be when all the destination nodes in U use the same path. In this case,

$$\Phi_{min} = \sum_{e \in p_i} \varsigma(e) \cdot F(U) = l_{avg}.\varsigma_{avg}.F(U) \tag{6.27}$$

From Eq.6.26 and Eq.6.27,

$$\Phi_{max} - \Phi_{min} = l_{avg}.\varsigma_{avg}(n - F(U)) \tag{6.28}$$

Here, $|U| = n$ that implies $F(U) \approx lg(n)$. Hence $n >> F(U)$. If destination nodes i updates its path from p_i to p'_i under the PSA, then

$$\Phi(p'_i, p_i) + \delta_i \leq \Phi(p_i, p_i) \tag{6.29}$$

Where δ_i is the decrement in the potential function. If destination node i leaves at least one edge while changing the path and shares one edge with another destination node, then the change in potential function will be at least $\varsigma_{min} := \min(\varsigma(e))$. Thus,

$$\Phi(p'_i, p_i) + \varsigma_{min} \leq \Phi(p_i, p_i) \tag{6.30}$$

The potential function value at Nash equilibrium will always be exist between the Φ_{max}

and Φ_{min}. Hence from Eq.6.28 and Eq.6.30 the possible number of iteration to achieve a Nash equilibrium will be $Q \leq n.r_{max}$.

6.6 Theoretical Evaluation

This section theoretically evaluates the performance of the PSA in terms of the overall cost of the multicast tree. Under the proposed cost-sharing scheme, the PSA aims to find a multicast tree as the PNE that minimizes the cost of each destination node discussed in Section 6.3.2. It behaves like a gradient descent process to find the solution by exploring all the possibilities. The optimum stable solution is called the best PNE. The quality of the best PNE is measured in terms of price of stability (PoS), which measures the ratio between the best PNE and centralized optimum. To obtain the bound for PoS, we first establish a relation between the total cost and the potential function.

Lemma 6.1 *For any action profile p, $\zeta(p) \leq \Phi(p) \leq F(U).\zeta(p)$.*

Proof: Given a destination node-set $U = \{1, \cdots, n\}$ and path selection profile $p = (p_1, \cdots, p_n)$, the multicast network formed by all paths in p from source to destination nodes is represented as $N(V', E') = \cup_i^n p_i$. Thus, the total cost of a multicast network for all the destination nodes is,

$$\zeta(p) = \sum_{e \in E'} \varsigma(e) \tag{6.31}$$

First, for each edge $e \in E'$, $F(S_e) \geq 1$ from Eq.6.19 as all the edges in E' are used by at least one destination node. Thus,

$$\sum_{e \in E'} \varsigma(e) \leq \sum_{e \in E'} \varsigma(e) \cdot F(S_e) \tag{6.32}$$

From Eq.6.31, Eq.6.18,and Eq.6.32

$$\zeta(p) \leq \Phi(p) \tag{6.33}$$

Second, for every edge $e \in E'$, $F(S_e) \leq F(U)$ because destination user node set $S_e \subseteq U$.

$$\sum_{e \in E'} \varsigma(e) \cdot F(S_e) \leq \sum_{e \in E'} \varsigma(e) \cdot F(U) \Rightarrow \Phi(p) \leq F(U) \cdot \zeta(p) \tag{6.34}$$

From Eq.6.33 and Eq.6.34, $\zeta(p) \leq \Phi(p) \leq F(U).\zeta(p)$. □

Theorem 6.3 *The price of stability for the identified cost-sharing scheme in the PS-Game with destination node-set U is at most $F(U)$.*

Proof: Let $\zeta(p^o)$ be the centralized optimum cost of all destinations for multicast at path selection profile p^o. Suppose the iterative process of the proposed PSA to find PNE starts with p^o. Let system reach the PNE at path selection profile p^* and total cost at p^* is $\zeta(p^*)$. As in Theorem 6.1, the value of the potential function decreases with every iteration of the PSA. Hence,

$$\Phi(p) \leq \Phi(p^*) \tag{6.35}$$

From Lemma 6.1,

$$\zeta(p) \leq \Phi(p) \text{ and } \Phi(p^*) \leq F(U)\zeta(p^*) \tag{6.36}$$

From inequalities in Eq.6.35 and Eq.6.36,

$$\zeta(p) \leq F(U) \cdot \zeta(p^*) \Rightarrow \frac{\zeta(p)}{\zeta(p^*)} \leq F(U) \cdot$$

Hence, the price of stability for PSGame is at most $F(U)$. □

Theorem 6.4 *The upper bound on PoS in the PSGame with set U of n destination nodes is always less than equal to $O(log(n))$.*

Proof: The harmonic series $H(n) = \sum_{i=1}^{n}(\frac{1}{i}) = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} = O(log(n))$. For proposed algorithm, the upper bound on PoS as proved in Theorem 6.3 is,

$$F(U) = \sum_{i \in U} \frac{c_i(p^*)}{\sum_{k \in U, c_k(p^*) \geq c_i(p^*)} c_k(p^*)} \tag{6.37}$$

The order of destination nodes in U according to their cost function at the PNE p^* is as follows:

$$c_{a_1}(p^*) \geq c_{a_2}(p^*) \geq c_{a_3}(p^*) \geq \cdots \geq c_{a_n}(p^*) \tag{6.38}$$

Where any $a_j \in U$. By expanding the $F(U)$ according to Eq.6.37 and Eq.6.38, we get,

$$F(U) = \frac{c_{a_1}(p^*)}{c_{a_1}(p^*)} + \frac{c_{a_2}(p^*)}{c_{a_1}(p^*) + c_{a_2}(p^*)} + \cdots + \frac{c_{a_n}(p^*)}{c_{a_1}(p^*) + c_{a_2}(p^*) + \cdots + c_{a_n}(p^*)}$$

$$H(n) - F(U) = \left(1 - \frac{c_{a_1}(p^*)}{c_{a_1}(p^*)}\right) + \left(\frac{1}{2} - \frac{c_{a_2}(p^*)}{c_{a_1}(p^*) + c_{a_2}(p^*)}\right) + \cdots + \left(\frac{1}{n} - \frac{c_{a_n}(p^*)}{c_{a_1}(p^*) + c_{a_2}(p^*) + \cdots + c_{a_n}(p^*)}\right)$$

As stated earlier, for any j^{th} term in $F(U)$, such that $1 \leq j \leq n$; $c_{a_1}(p^*), c_{a_2}(p^*), \cdots, c_{a_j}(p^*) \geq c_{a_j}(p^*)$. Therefore $\left(\frac{1}{j} - \frac{c_{a_j}(p^*)}{c_{a_1}(p^*)+c_{a_2}(p^*)+\cdots+c_{a_j}(p^*)}\right) \geq \left(\frac{1}{j} - \frac{c_{a_j}(p^*)}{j \cdot c_{a_j}(p^*)}\right) \geq 0$. It implies that $H(n) - F(U) \geq 0$, or $F(U) \leq H(n)$. Hence, the PoS for the proposed PSA is never reached more than $O(log(n))$. □

6.7 Illustration

This section presents two examples that give numerical results to validate the theoretical results, as provided in Fig. 6.4 and Fig. 6.5. The first example shows the scenario of PSGame with two destination nodes, and the second example represents the scenario with n nodes. In both cases, we prove that there will be at least one PNE, and the PoS

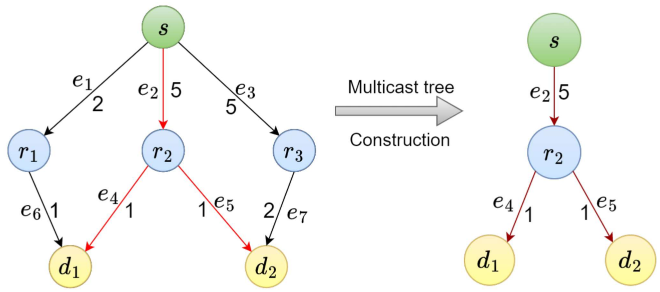


Figure 6.4: Example of a network with two destination nodes and a PNE as a multicast tree.

is less than $O(logn)$.

6.7.1 Example 1: Two Destination Nodes

In this example, a source node s feeds the destination nodes d_1 and d_2, as shown in Fig. 6.4. As described above in Section 6.2, the total cost of all destination nodes with the Shapley cost-sharing mechanism at PNE is 9, while the total cost of the global optimum solution is 7. It observes that the unique PNE with Shapley cost-sharing mechanism differs from the centralized optimum.

PSGame: Let the destination nodes $< d_1, d_2 >$ begin with an arbitrary path selection profile $p = (\{e_1, e_6\}, \{e_3, e_7\})$ and initial weights $(1, 1)$. The cost of d_1 and d_2 at this path selection profile will be 3 and 7, respectively, from Eq.6.13. At the path selection profile $(\{e_1, e_6\}, \{e_3, e_7\})$, destination node d_2's path cost is not a minimum. Therefore, it can change its path from $\{e_3, e_7\}$ to $\{e_2, e_5\}$. As a result, the new path selection profile will be $(\{e_1, e_6\}, \{e_2, e_5\})$, and the cost incurred by d_1 and d_2 at this path selection profile will be 3 and 5, respectively, from Eq.6.15. Now, destination node d_1 may also improve its cost by changing the path from $(\{e_1, e_6\}$ to $\{e_2, e_4\}$. The costs incurred by d_1 and d_2 at path selection profile $(\{e_2, e_4\}, \{e_2, e_5\})$ will be 2.67 and 3.33, respectively, from Eq.6.15. It will be final path selection profile that construct an stable multicast tree for group communication.

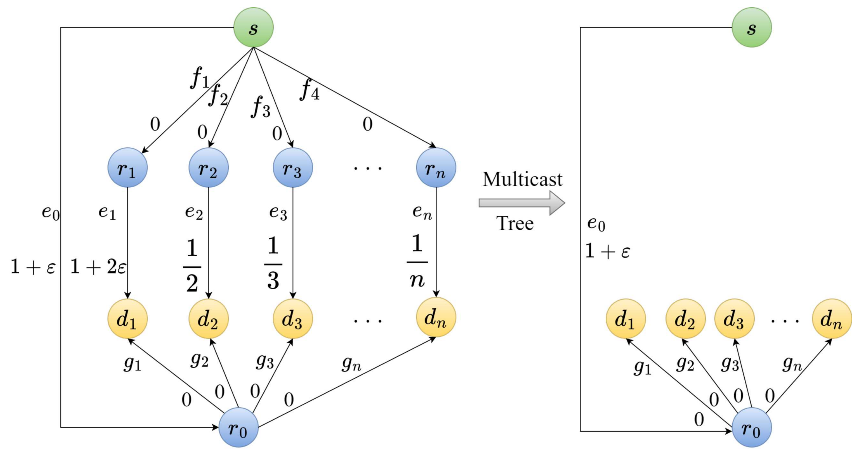


Figure 6.5: Example of a network with n destination nodes and a PNE as a multicast tree.

Pure Nash Equilibrium: The path selection profile $a^*(\{e_2, e_4\}, \{e_2, e_5\})$ is the PNE and also an optimum. At this path selection profile, no destination node can reduce its cost by changing its path unilaterally.

Price of Stability: The overall cost at the PNE and centralized optimum are same as 7. As a result, the PoS is,

$$PoS(a^*) = \frac{\text{Overall cost at PNE}}{\text{Overall cost at central optimum}} = 1$$

Hence, the PNE solution admitted in the PSGame for this 2 destination node example is optimum.

6.7.2 Example 2: n Destination Nodes

In this example, a source node provides the feeds to n destination nodes, as shown in Fig. 6.5. Suppose destination node d_i has its own path of cost $\frac{1}{i}$, as given in Fig. 6.5. Initially, if all destination nodes share a common path of cost $1 + \varepsilon$ for some small $\varepsilon > 0$, which is also a centralized optimum. The optimal solution would connect all destination nodes through the common path for a total cost of $1 + \varepsilon$. However, if this solution were offered to the users with Shapley cost-sharing mechanism, except d_1 they would defect from it one by one to their alternate paths. The system has a cost of $\sum_{i=1}^{n} \frac{1}{i} = H(n)$.

PSGame: According to our proposed weighted cost-sharing scheme, initially, destination nodes may defect from the centralized optimum solution because each destination node's default weight is 1. But as their weights change with their path update, all the destination nodes will start to share the common edge with cost $1 + \varepsilon$. Suppose, their initial path selection is $(\{f_1, e_1\}, \{f_2, e_2\}, \{f_3, e_3\}, \cdots, \{f_n, e_n\})$. Now, destination node d_1 deviates from e_1 to e. Then, node d_2 deviates from e_2 to e as share the cost according to Eq. 6.15. Similarly, each node will deviate to the common edge while sharing the edge cost according to Eq. 6.15. Thus, the final path selection profile will be $(\{e_0, g_1\}, \{e_0, g_2\}, \{e_0, g_3\}, \cdots \{e_0, g_n\}$, which construct an stable multicast tree for group communication.

Pure Nash Equilibrium: The cost incurred by any destination node d_i will be less than $\frac{1}{i}$. Therefore, no destination node will unilaterally deviates from the path selection profile $a^* = (\{e_0, g_1\}, \{e_0, g_2\}, \{e_0, g_3\}, \cdots \{e_0, g_n\})$.

Price of Stability: The overall cost at the centralized solution and PNE are same $(1 + \varepsilon)$, as a result the PoS is,

$$PoS(a^*) = \frac{\text{Overall cost at PNE}}{\text{Overall cost at central optimum}} = \frac{1+\varepsilon}{1+\varepsilon} = 1$$

Hence, the PNE solution admitted in the PSGame for this n destination nodes example is optimum.

6.8 Simulations and Results

This section presents a numerical analysis of the proposed PSA. The analysis exhibits the impact of different network properties on the cost, delay, packet loss, jitter, throughput, and energy consumption. We use the NS2 simulator to implement our proposed algorithm. The simulation is conducted on a Ubuntu 20.04.3 LTS machine equipped with an Intel Core i5-7400T processor (4 CPUs, 2.4GHz) and 8GB RAM. The PSA is compared to state-of-the-art approaches for finding the multicast tree for group communication, namely Particle Swarm Optimization (PSO) [31], Destination-oriented Zone Squirrel Search Algorithm (DZSSA) [53], Game-Theoretic Algorithm with Shapley cost-sharing (GTAS) [112], and Grey Wolf Optimization (GWO) [145], which are described as follows:

- PSO: A meta-heuristic approach searches for an optimal solution in the solution space. The PSO iteratively optimizes the multicast tree by improving it.
- DZSSA: This meta-heuristic approach is an improved version of the squirrel search algorithm (SSA). This approach finds an optimal multicast tree by acting in the destination-oriented zones (DZ).
- GTAS: It is a decentralized approach that gives the decision-making capabilities to the destination nodes for constructing the cost-effective multicast tree. This approach used the Shapley cost-sharing mechanism to divide the cost of an edge among the edge's users.
- GWO: A meta-heuristic-based approach divides the underlying networks into the optimized number of clusters. These clusters lead to a robust routing protocol for group communication.

Table 6.1: Networks Topology

	Network Size (nodes)	Multicast Group Size	Transmission Range of nodes (in meters)	Source Nodes
Set#1	300	50, ..., 130	60	1
Set#2	300	60	50, ..., 90	1
Set#3	100, ..., 500	50	70	1

Table 6.2: Simulation Parameters

Name	Description
Interface Queue Type	Droptail/ PriQueue
Queue Length	50 Packets
Channel Type	WirelessChannel
Propagation Delay	ConstantSpeedPropagationDelay
Propagation Loss	FriisPropagationLossModel
Antenna	OmniAntenna
MAC Layer	IEEE 802.15.4 MAC (CSMA/CA)
Link Layer Type	LL
Physical Layer	IEEE 802.15.4 PHY
Node Speed	UniformRandomSpeedModel (0 to 20 m/s)
Node Movement	RandomWayPointMobilityModel
Simulation Time	500s
Data Packet Size	100 Bytes
Broadcast Packet Size	32 Bytes
Initial Battery Power	200 Joule
ξ_{td}	0.3 μW
ξ_{fa}, ξ_{ae}	0.4 μW
$(\kappa_1, \kappa_2, \kappa_3)$	(0.6, 0.2, 0.2)
Link's Bandwidth	$\{1 \text{ Mbps}, 2 \text{ Mbps}, \cdots, 5 \text{ Mbps}\}$ uniformaly randomly assigned

6.8.1 Simulation Setting

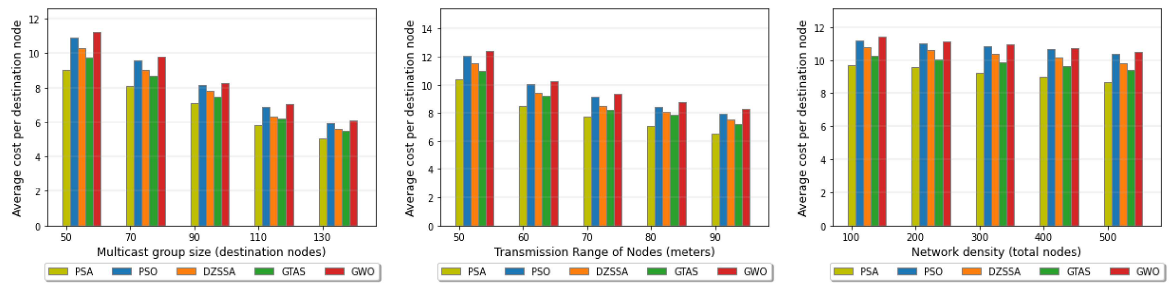
6.8.1.1 Network Topology

The experiments are conducted on the real IoT user datasets [2], which contains the geographical locations of IoT device users, and cellular base stations in Australia. Simulations are carried out for an highly dense urban area of 500 $m \times 1000$ m. In order to evaluate the algorithm's performance in a variety of environments and get the rich analysis, we obtain the results by varying the parameters of the network topology, e.g., multicast group size, transmission range of nodes, and network density (total nodes), as given in Table 6.1. To vary the network density, the geographical location coordinates of IoT users are selected uniformly at random from the set of all points in the simulated area as nodes, and some of these chosen nodes are randomly selected as a multicast group (destination nodes). One of the cellular base stations in the simulated area is randomly selected as the source node which is considered as the control server. The obtained networks (Table 6.1) by altering attributes of the network topology in the simulated area are described as follows:

- Set#1: The networks are obtained by varying destination nodes from 50 to 130. It observes the effect of the size of the multicast group on the performance of the PSA.
- Set#2: The networks are obtained by varying the transmission range of the nodes (destination and intermediate) from 50 m to 90 m. Increasing the range of nodes means that any node can communicate directly with more nodes than before, increasing the edges in the network.
- Set#3: The networks are obtained by varying the number of nodes chosen from the simulated area from 100 to 500. It measures the proposed algorithm's performance on a scale from sparse to dense network.

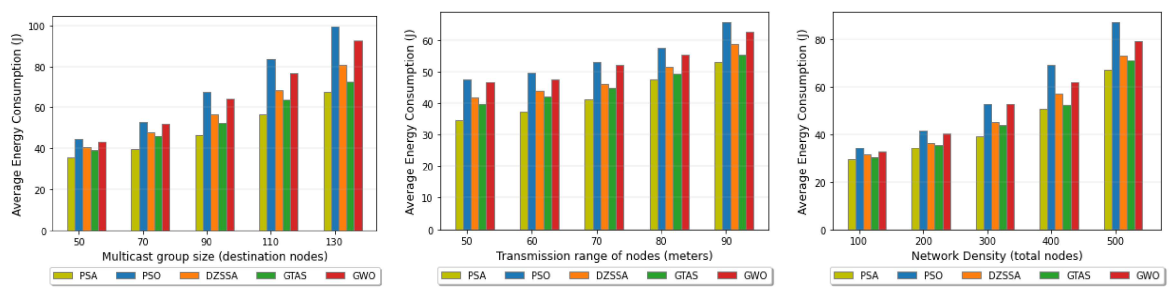
The range of the source node is fixed at 100 m in all the experiments. In each

[2] https://github.com/swinedge/eua-dataset



(a) (b) (c)

Figure 6.6: Cost comparison of constructed multicast tree for group communication. (a) Set#1. (b) Set#2. (c) Set#3.



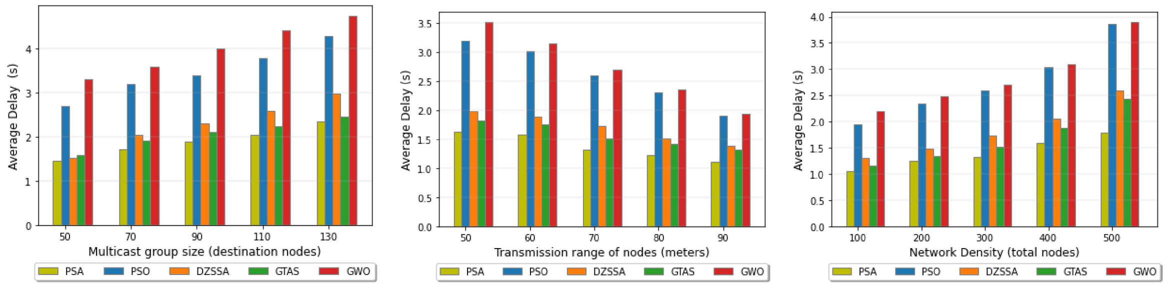
(a) (b) (c)

Figure 6.7: Performance evaluation based on the energy consumption in the group communication. (a) Set#1. (b) Set#2. (c) Set#3.

experiment, a multicast tree is first constructed for group communication, and the traffic is then measured between the source node and the destination nodes.

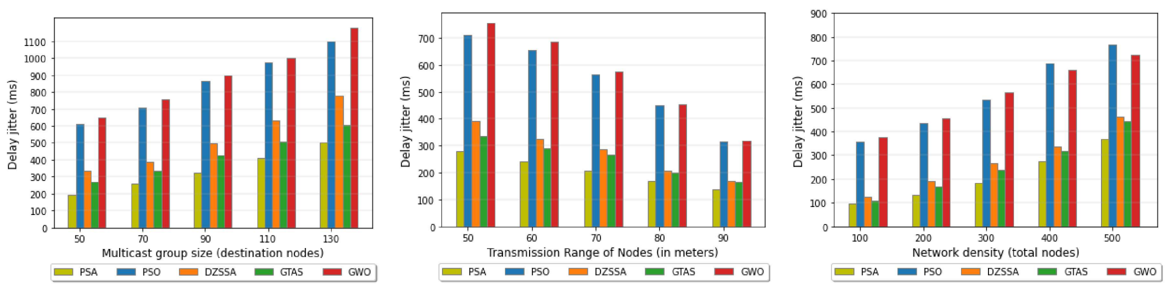
6.8.1.2 Parameter Settings

We deploy the proposed PSA under the NS2 simulator on top of the existing IEEE 802.15.4 implementation as in [146]. For each experiment, the bandwidth to each link between any two nodes is uniformly randomly assigned from set $\{1 \text{ Mbps}, 2 \text{ Mbps}, \cdots, 5 \text{ Mbps}\}$, and the reference bandwidth is 10 Mbps. Initially, each node has a battery with 200 Joule energy [147]. The length of the broadcast packets (used in constructing the multicast tree) and data packets are 32 bytes and 100 bytes, respectively. The transmission range of nodes is given in Table 6.1. The power consumption ξ_{td} by the transceiver circuit to transmit or receive a single bit is set to 0.3 μW. The power



(a) (b) (c)

Figure 6.8: Performance evaluation based on the average delay in the group communication. (a) Set#1. (b) Set#2. (c) Set#3.



(a) (b) (c)

Figure 6.9: Comparison of the jitter in group communication through a constructed multicast tree. (a) Set#1. (b) Set#2. (c) Set#3.

consumption ξ_{fa} by the amplifier circuit to amplify a single bit is set to 0.4 μW. The energy consumption in transmitting or receiving a packets is calculated by the Eq. 6.5. The cost of each edge between any two nodes is derived from Eq. 6.1. We are giving more weightage to the throughput ($\kappa_1 = 0.6$) than delay ($\kappa_2 = 0.2$) and energy consumption ($\kappa_3 = 0.2$) as we model the throughput-intensive applications. These weights can be adjusted according to the application's requirements. The maximum length of each interface queue at every node is 50. We repeat each experiment 50 times, and the results were then averaged. This neutralizes extreme cases, such as overly dense or sparse nodes distributions. For simplicity, Table 6.2 summarizes the simulation parameters.

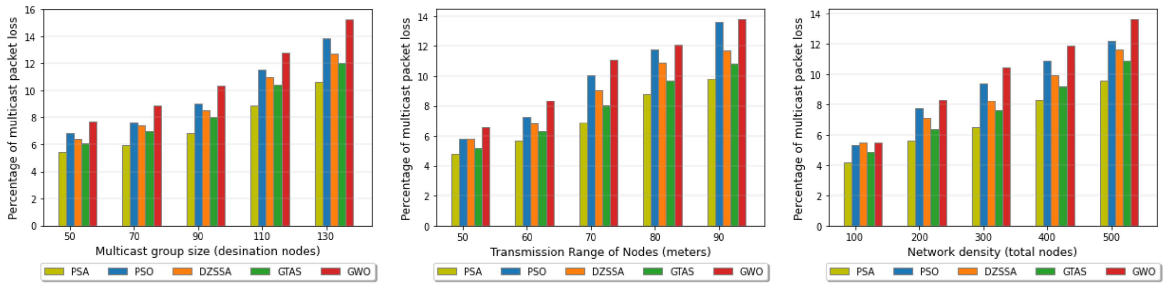
6.8.2 Performance Analysis

To observe the effectiveness of the PSA, we compare it with state-of-the-art approaches based on six metrics as follows:

1. Cost: It represents the average path cost incurred by each destination nodes in the multicast group, which is the ratio of the multicast tree cost to the number of destination nodes.
2. Delay: It contains two components: end-to-end delay and per packet multicast tree construction overhead. End-to-end delay is the average time for data packets to reach the destination. Multicast tree construction overhead for each packet is the ratio between the time required to construct a multicast tree and the number of packets received by the destination.
3. Jitter: It refers to the variation in the end-to-end delay.
4. Packet loss: It is the overall packet loss between the source to destinations in the network.
5. Throughput: We measure the average data rate received by each destination node as the throughput.
6. Energy consumption: We consider the average energy consumption by each node in the multicast tree construction and the data packet transmission.

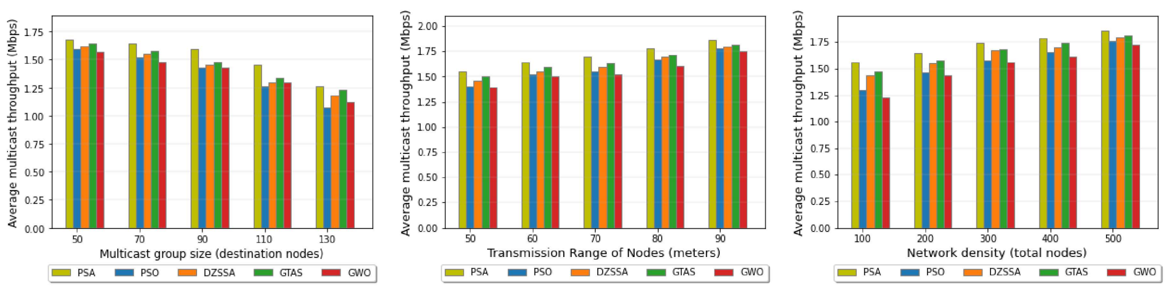
6.8.2.1 Cost Analysis

The results depicted in Fig. 6.6 represents the average path cost of each destination node. The results illustrated in Fig. 6.6a show that the average cost of destination nodes is inversely proportional to the number of destination nodes. The reason is that an increasing number of destination nodes results in sharing the same edge among more nodes than previously. Fig. 6.6b shows the effect of increasing the transmission range of nodes on the average cost of destination nodes. An increase in transmission range shortens the paths' length in terms of edges. Additionally, an increase in transmission



(a) (b) (c)

Figure 6.10: Performance evaluation based on the packet loss in the group communication. (a) Set#1. (b) Set#2. (c) Set#3.



(a) (b) (c)

Figure 6.11: Performance evaluation based on the throughput in the group communication. (a) Set#1. (b) Set#2. (c) Set#3.

range increases the number of paths between the source and destinations, which gives more choices to select a multicast tree with lower cost. Due to these reasons, the average cost decreases when the transmission range increases. As shown in Fig. 6.6c, if the network density increases, the average cost of the destination nodes decreases slightly. This happens because the total number of possible paths between the source and destinations increases, leading to a more cost-effective multicast tree, as aforementioned. These results show that the PSA performs well in each case. It is due to the fact that PSA motivates the destination nodes to share the edges at a higher degree that maximizes resource utilization. The optimization function used in PSA prioritizes edges with higher bandwidth, lower delay, and lower energy consumption. In other words, the PSA constructs a cost-effective multicast tree with overall higher bandwidth, lower delay, and lower energy consumption than other algorithms.

6.8.2.2 Energy Consumption Analysis

The results illustrated in Figs. 6.7a-6.7c show that with the increase in the multicast group size, transmission range, and network density, the overhead associated with the multicast tree construction will also increase in all the algorithms. This is because the number of paths among the nodes also increases with these topology parameters. Fig. 6.7 shows that PSA's energy consumption is always less than other algorithms. This happens as it reduces the search space and complexity for finding the solution by giving decision-making capability to each destination node. Additionally, any destination node only participates in the path update process when path cost is reduced. Furthermore, the group communication through the multicast tree constructed by PSA requires lower packet retransmission. PSA also optimizes the energy consumption needed for group communication by considering the energy consumption as one of the parameters in the cost. As a result, the PSA consumes less energy during multicast data transmission.

6.8.2.3 Delay Analysis

Fig. 6.8 shows that the delay for the proposed PSA is less compared to other state-of-the-art approaches. The proposed PSA effectively deals with the loop formation problem because each destination node constructs its path independently. At the same time, different conventional approaches requires high computations to deal with loop formation in multicast routing with multiple objectives that cause the overhead. Moreover, finding an effective solution using PSA is less complex than other algorithms, reducing the delay. Another reason for better performance is that it creates an optimal multicast tree with high-bandwidth by increasing the edge-sharing that minimizes the transmission delay. Fig. 6.8a shows that the average delay in each approach increases with the number of destination nodes as finding a cost-effective solution becomes complex and also increases the multicast tree size. The results illustrated in Fig. 6.8b show the average delay reduces with increasing transmission range. The reason is that an increased transmission range reduces the number of data forwarding nodes in the multicast tree that lower the transmission and queuing delay. As shown in Fig. 6.8c, the delay increases as the total number of nodes in a given geographic area increases. This happens as the increase in network density also increases the network congestion.

6.8.2.4 Jitter Analysis

The results illustrated in Fig. 6.9 show the jitter against the different network topology parameters. The PSA lowers the random waiting time spent in the queue required to avoid collisions that minimizes the jitter. The reason for this is that PSA allocates paths with higher bandwidth to reduce the transmission time, which minimizes the vulnerable time of the packet collision. Thus, the PSA performs better than other approaches. If the multicast group size is increased, the size of the constructed multicast tree is also increased. This leads to the possibility of including congested links, which increases the network jitter for each approach, as shown in Fig. 6.9a. Similarly, an increase in the network density in a given geographical area increases the network congestion, which results in an unpredictable delay, as shown in Fig. 6.9c. On the other hand, the results illustrated in Fig. 6.9b shows that the jitter reduces with the increasing transmission range of nodes. This happens as increasing transmission range reduces the number of data forwarding nodes in the multicast tree.

6.8.2.5 Packet Loss Analysis

The results shown in Fig. 6.10 represent the end-to-end packet loss in the network for group communication. If the size of any parameter (multicast group size, transmission range, and network density) is increased, the chances of packet collision will also increase during the group communication. Therefore, the packet loss increases in all the approaches with increment in any of the network topology parameter, as shown in Fig. 6.10a-6.10c. The multicast tree constructed by PSA has more common edges than other algorithms. As a result, overall packet losses are lower due to propagation loss. The PSA also reduces the packet drop rate at intermediate nodes by keeping the queue lengths under control. The reason is that it balances the incoming and outgoing packet rates using the higher bandwidth multicast tree. Thus, the end-to-end multicast packets loss is lower in our proposed algorithm.

6.8.2.6 Throughput Analysis

The results depicted in Fig. 6.11 show the average end-to-end multicast throughput for group communication. The PSA supports high throughput because destination nodes only switch to the alternate path if it is cost-effective in terms of bandwidth. If the multicast group size increases, the multicast tree size for group communication will also increase. It can lead to including some overloaded edges in the multicast tree. Therefore, the throughput decreases with increasing multicast group size, as shown in Fig. 6.11a. The higher transmission range of nodes increases the possible choices of multicast trees, enabling the selection of the best tree with higher bandwidth. Consequently, it increases throughput. However, throughput increases by a small amount as the increased transmission range also increases the chances of packet collisions, as shown in Fig. 6.11b. Similarly, the results in Fig. 6.11c show that throughput increases with network density.

6.9 Summary

This chapter proposed a PSGame to construct a multicast tree that minimizes the overall cost of IoT devices equipped with sensors for group communication. We designed a path selection algorithm (PSA) to converge at Nash equilibrium by following PSGame rules. This algorithm uses a proposed cost-sharing scheme to improve the quality of the Nash equilibrium. We proved that the proposed PSA reaches at least one Nash equilibrium. The time-bound for the convergence at the Nash equilibrium is $O(n.r_{max})$.

The proposed algorithm's performance is analyzed theoretically, proving that the price of stability for the obtained solution will always be less than $O(log(n))$. We also conducted experiments to numerically evaluate the algorithm's performance and compared the results with other game-theoretic, heuristic, and meta-heuristic algorithms. Results show that the proposed algorithm constructs the multicast tree of lesser cost with minimal overhead than the other state-of-the-art algorithms.