Energy Efficient IoT Networks

GreenEdge Contest

K. Botirov, J. Fernandez Landivar, SA. Lahmer, & L. Scalambrin

Introduction to The Challenge

The goal of this challenge is to solve an optimization problem in the context of multi-hop networks, commonly observed in scenarios where IoT is utilized for monitoring extensive areas. Fig. 1 illustrates a possible scenario where **sensor nodes** collect data that is later transmitted to the **gateway nodes**, which are the ones with internet access, through intermediary **relay nodes** strategically placed. Such optimization results from the maximization of a function $\widetilde{G}$, which is a linear combination of the Network Throughput T, Energy Efficiency η and Network Utilization $\mathcal{U}$ objectives. All these objectives and their corresponding constraints are defined and explained in the following sections. At the end of this document, you will find the format expected for the solutions, together with an example code and some tools that can be used to solve the challenge.

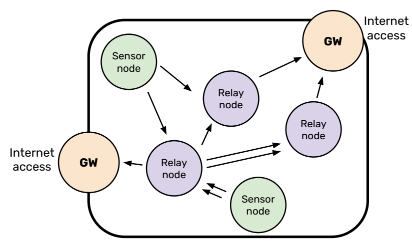


Figure 1: Possible configuration of gateway, sensor, and relay nodes.

Problem Definition and Notation

Table 1 comprises the problem definition and the notation for the challenge. It summarizes the characteristics of the IoT network's nodes and edges. The table also provides information about the different size limits.

Objectives

The overall objective consists of optimizing a function that combines multiple specific target performance indexes, as defined below.

1. Network Throughput (T)

Maximize the total data successfully delivered to the gateways:

$$T = \sum_{s \in S} \sum_{g \in G} D_{s_g}$$

Each source node s must have assigned only one gateway as a destination.

Variable	Description	Values	Observations
$G = \{g_1, g_2, \ldots, g_m\}$	A set of gateway nodes	1 to 100	Gateways receive data from the source nodes
$S = \{s_1, s_2, \ldots, s_k\}$	A set of source nodes	1 to 10000	Sources are characterized by their maximum bandwidth required
$R = \{r_1, r_2, \ldots, r_l\}$	A set of relay nodes	0 to 10000	Relay nodes retransmit data
$V = \{v_1, v_2, \ldots, v_N\}$	A set of all network nodes	1 to 10000	Where $V = G \cup S \cup R$ and $G \cap S \cap R = \emptyset$
$E = \{(v_i, v_j) \| v_i, v_j \in V\}$	A set of edges between nodes	Depends on the number of nodes.	The edges are characterized by their maximum data rate capacity C_{ij} and average transmission power P_{ij}. There may be 1 up to 2 edges connecting two nodes $v_i, v_j \in V$.
D_{s_g}	Data flow from source node s successfully delivered to gateway g.		D_{s_g} can be equal to or less than the bandwidth required by the sources, depending on the network capacity
$f_{s_g}(v_i, v_j)$	Data flow on link (v_i, v_j) from source node s to gateway node g		

Table 1: Principal variables and notation

Explanation: Focus on maximizing the aggregate of data transferred from all sources to all gateways. It can be seen as a Max-Flow problem.

2. Energy Considerations

a. Energy Consumption ($\mathcal{E}$)

The total energy consumption in the network for transmitting data:

$$\mathcal{E} = \sum_{(v_i, v_j) \in E} P_{ij} \times \tau_{ij} \times \sum_{s \in S} \sum_{g \in G} f_{s_g}(v_i, v_j)$$

where

$$\tau_{ij} = \frac{1}{C_{ij}}$$

Explanation: Aim to characterize the total energy spent in transmitting data across the network.

b. Energy Efficiency (η)

The energy efficiency, defined as the total data delivered per unit of energy consumed:

$$\eta = \frac{\sum_{s \in S} \sum_{g \in G} D_{s_g}}{\mathcal{E}}$$

Explanation: Attempt to optimize the ratio of the total data delivered to the total energy used (energy per bit transmission). This is about delivering more data while consuming less energy.

3. Network Utilization ($\mathcal{U}$)

Network utilization for a given link is defined as the ratio of the actual data flow through the link to its maximum capacity. A link is considered utilized if there is at least one data flow passing through it. The objective is to ensure that the network links are efficiently utilized.

For a link (v_i, v_j) in E, its utilization $\mathcal{U}$ is given by:

$$\mathcal{U}(v_i, v_j) = \frac{\sum_{s \in S} \sum_{g \in G} f_{s_g}(v_i, v_j)}{C_{ij}}$$

where C_{ij} is the maximum capacity of the link (v_i, v_j).

The overall network utilization objective can be defined as:

$$\text{Maximize } \mathcal{Z}_U = \sum_{(v_i, v_j) \in E} \mathcal{U}(v_i, v_j)$$

Such that:

$$0 \leq \mathcal{U}(v_i, v_j) \leq 1, \quad \forall (v_i, v_j) \in E \tag{1}$$

Explanation: Focus on maximizing the utilization of each network link, ensuring that the network's capacity is used effectively and not wasted.

Global Objective

The global objective $\widetilde{G}$ is a linear combination of all sub-objectives normalized, weighted by coefficients representing the importance of each objective. Given coefficients α, β and δ:

$$\textbf{Maximize } \widetilde{G} = \alpha \times \eta + \beta \times T_{norm} + \delta \times \mathcal{Z}_{U_{norm}} \tag{2}$$

Explanation: The overall goal is to find the best balance between these objectives, considering their relative importance. The coefficients (α, β, δ) for each objective are not fixed and will be provided along with the problem input. This holistic approach ensures that the network is optimized for throughput, energy efficiency, and utilization simultaneously.

Constraints

Capacity Limit

The flow of any edge cannot exceed its capacity, in other words:

$$\sum_{s \in S} \sum_{g \in G} f_{s_g}(v_i, v_j) \leq C_{ij}, \quad \forall (v_i, v_j) \in E \tag{3}$$

Conservation of Flow

The principle of flow conservation dictates that the total inflow to a node must be equal to the total outflow from that node, with exceptions for source nodes where flow only exits, and gateway nodes where flow only enters. This can be mathematically expressed as follows:

$$\sum_{v_j \in V} f_{s_g}(v_i, v_j) - \sum_{v_k \in V} f_{s_g}(v_k, v_i) = \begin{cases} D_{s_g}, & \text{if } v_i = s \\ -D_{s_g}, & \text{if } v_i = g \\ 0, & \text{otherwise} \end{cases}, \quad \forall s \in S, \forall g \in G \tag{4}$$

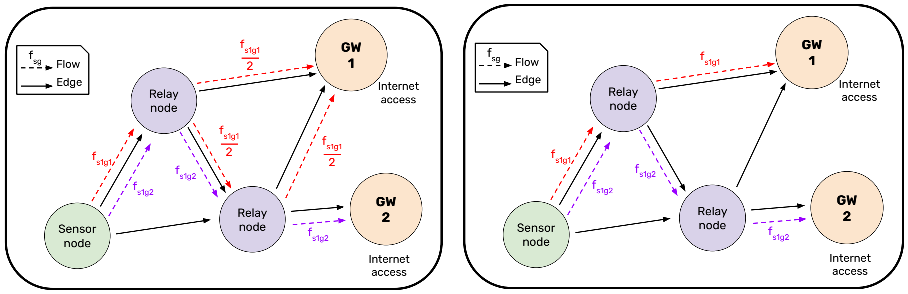
Path Consistency

Given that the problem allows for multiple sources and multiple sinks or destination nodes, it is essential to ensure that the flow from each source-sink pair $s - g$ remains undivided across different paths in the graph. Fig. 2 exhibits two flow distributions, 2a does not adhere to the path consistency constraint, while 2b demonstrates adherence to the constraint.

This requirement can be formally stated as follows:

$$f_{s_g}(v_i, v_j) = f_{s_g}(v_a, v_b), \quad \forall (v_i, v_j), (v_a, v_b) \in E, \quad \forall s \in S, \quad \forall g \in G \tag{5}$$

Explanation: Enforcing the path consistency constraint is crucial for maintaining efficient and reliable data flow in the network. Although protocols like TCP are capable of handling packet re-ordering relying on this feature excessively can overburden the end-hosts. This excessive load will lead to degrading the overall network Performance. Furthermore, for applications using UDP-like that do not naively handle packet ordering, path inconsistency can lead to errors in data processing, adversely affecting the overall application's performance and reliability.



(a) Invalid flow distribution.

(b) Valid flow distribution.

Figure 2: Scheme 2a illustrates an incorrect flow distribution where the flow $f_{s1_{g1}}$ is divided into two halves. In contrast, scheme 2b shows a valid configuration that follows the path consistency constraint, ensuring that $f_{s1_{g1}}$ and $f_{s1_{g2}}$ remain unified throughout the network.

Input/Output format

The solver is expected to read the input graph along with all parameters via stdin using the following format:

- The first line contains four integers: N, the number of nodes, k, the number of sources, m, the number of gateways, and $|E|$ the number of total edges.
- The next k lines contain IDs of the source nodes along with the requested bandwidth.
- The next line contains a list of GW node IDs.
- The next $|E|$ lines define edges, where each line contains five values: Edge ID, the ID of the starting node, the ID of the ending node, the capacity C_{ij}, and the transmission power P_{ij} of that link.
- The last line contains three floating point numbers representing the coefficients α, β, and δ, respectively, which weigh the importance of the objectives in the global objective function.

and then the solver should write the output to stdout in the following format:

- k lines, where each line contains a list of the edge IDs corresponding to the source flow path, followed by the flow value.

Example (see Fig. 3)

This section illustrates an example of a simple network scenario, as depicted in Fig. 3, which consists of one source, one gateway, and two relay nodes, to-

gether a sample input along with corresponding output, following the guidelines explained in the previous section.
Input:

```
4 1 1 6
0 10
3
0 0 1 10 3
1 0 1 5 3
2 0 2 15 2
3 1 3 10 2
4 1 2 20 1
5 2 3 10 3
0.33 0.33 0.33
```

Output:

```
0 3 10
```

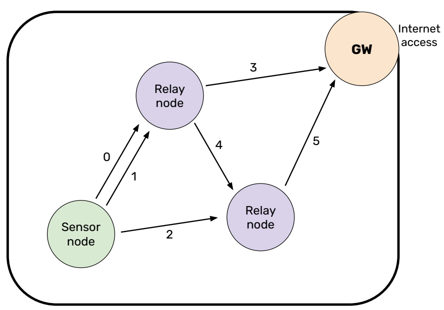


Figure 3: Example scenario

Code Example

As a guide, a code example of a baseline greedy algorithm for a simple solution in a network with one source, one gateway, and n relay nodes is presented in the following Jupyter notebook, implemented in Python. The solution's output is the path with a highest global objective score for the topology described. Further explanation is provided within the provided link.

Solution Requirements

The solution to the challenge can be carried out by an algorithm developed by the contestants on their own or by applying any solver/technique available, such as the library `pyomo` together with the solver `MindtPy`, for example. The algorithm's **execution time** must remain below one 1 second for any graph that complies with the maximum dimension values outlined in Table 1.

The final proposed solution will comprise a *ZIP* file that contains:

- The full code is developed in any programming language, following the input/output instructions mentioned above.
- A *PDF* document explaining the developed solution and the different approaches used to carry out this challenge, discussing the chosen strategy, libraries, solvers, and others (max 1 page).

Assessment and Evaluation Procedure

The set of all the solutions provided will be evaluated with a set of input graphs that comply with all the requirements described in the document (maximum number of nodes, maximum number of edges/links between connected nodes, etc).

The procedure to find the winning solution is as follows:

1. Input the graph into the i-th solution to solve the problem. Once the solution output is generated, it is evaluated to ensure that it qualifies as valid to participate, *i.e.*, capacity over the edges cannot be exceeded and the execution time must be within the established limits. If valid, the global objective $\widetilde{G}_i$ will be calculated and registered together with the execution time.
2. The first step is repeated for each solution across all graphs.
3. After calculating $\widetilde{G}_i$ for all solutions, $L-1$ points will be allocated to the solution with the highest $\widetilde{G}_i$, where L denotes the total number of contestants. Conversely, the solution with the lowest $\widetilde{G}_i$ will receive 0 points. If multiple solutions present identical $\widetilde{G}_i$ values, their ranking will be determined by their execution time; the solution with the shorter execution time will be awarded more points.
4. For every solution, the sum of all the points obtained over all the graphs will be computed. Then, the winner's solution is given by the one with the highest amount of points. If more than one solution presents the same amount of points, the solution with the shortest total execution time, *i.e.*, the sum of execution time across all graphs, is declared the winner.