Optimal Data Task Distribution for Balancing Energy Consumption on Cooperative Fog Networks

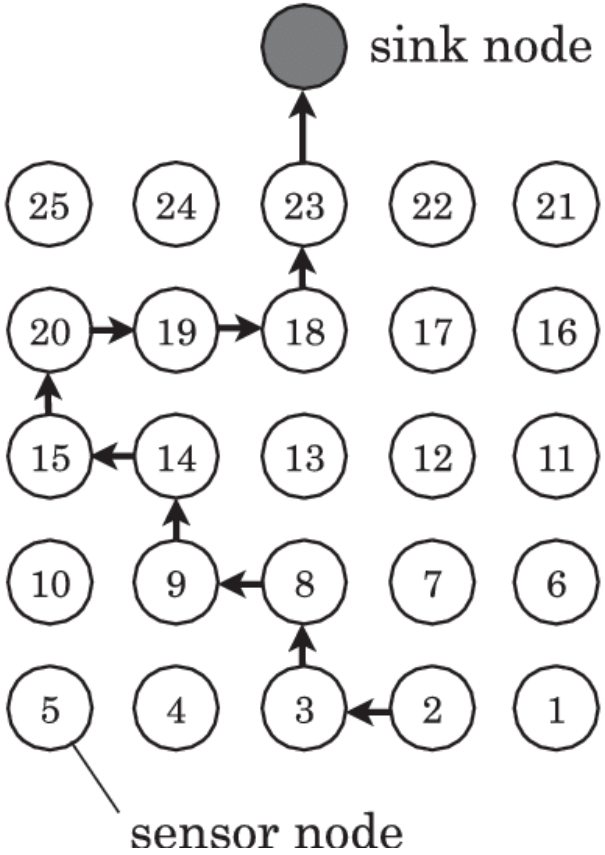
Jose Clemente, ***Fangyu Li**** *and WenZhan Song*
Center for Cyber-Physical Systems, University of Georgia

UNIVERSITY OF GEORGIA
College of Engineering

Outline

- Motivation and Contribution
- Network Model
- Graph Transformation and Formulation
- Algorithm
- Analysis and Evaluation
- Conclusion

Motivation

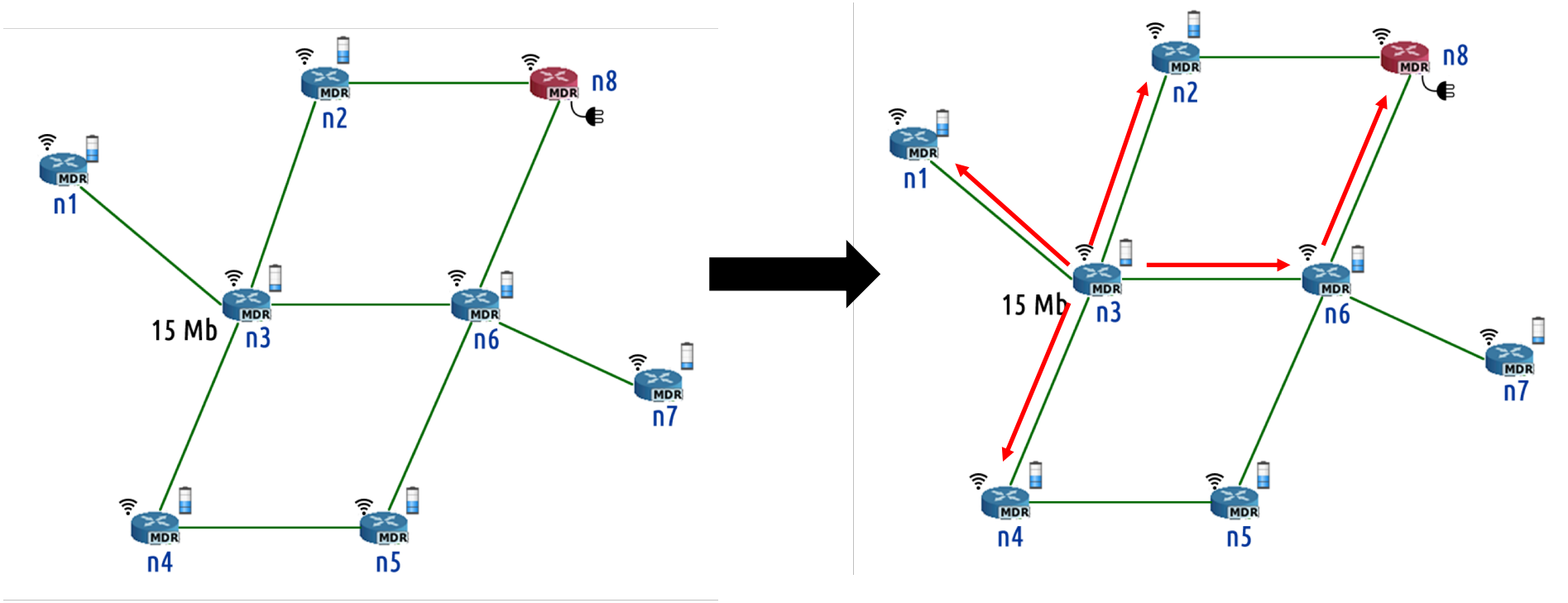


- In traditional sensor networks, data gathered by sensor nodes is usually forwarded to a sink node for post-processing.
- An unbalanced energy consumption will decrease the lifetime of the nodes (mostly in stick nodes).
- Some routing schemes attempt to find the minimum energy path to the sink to optimize energy usage at nodes [5-7].

Motivation

- Fog computing paradigm brings the data processing, networking, storage and analytics closer to the devices, and applications.
- In a fog network where nodes attempt to help other nodes by computing part of their information, balancing energy become a more complex problem.

Motivation

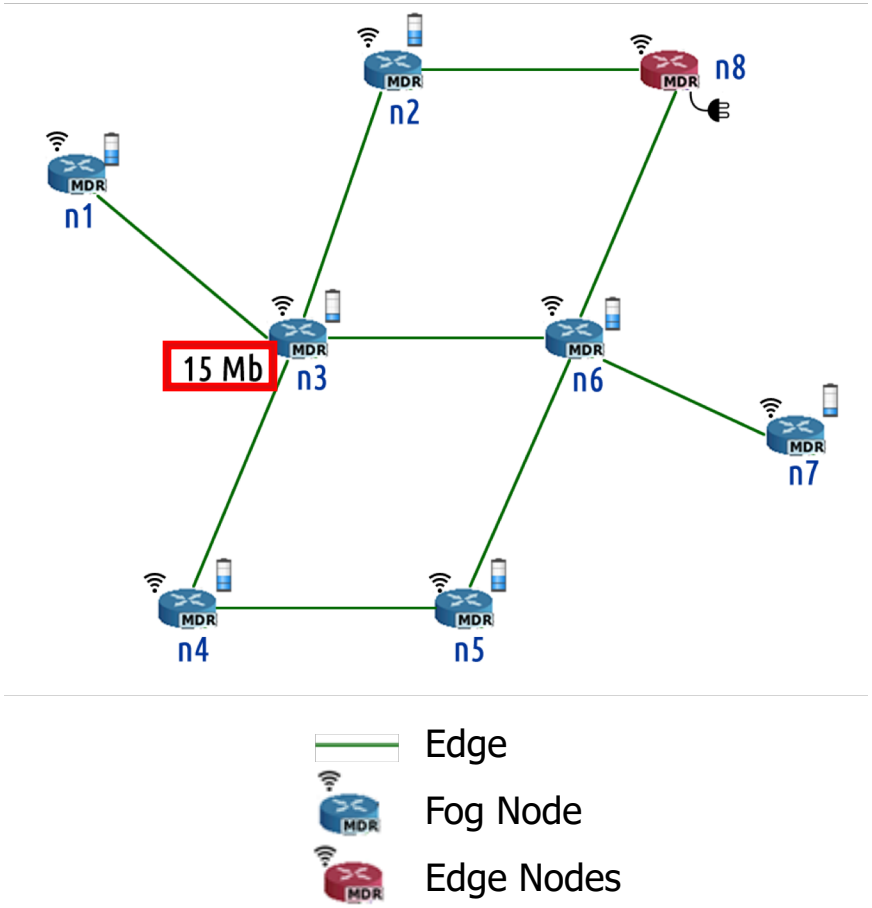


- Node **n3** may split its data to enable two or more nodes to help it in the processing. However, choosing the nodes is not trivial and has a critical impact on the power balance of the fog network.

Contribution

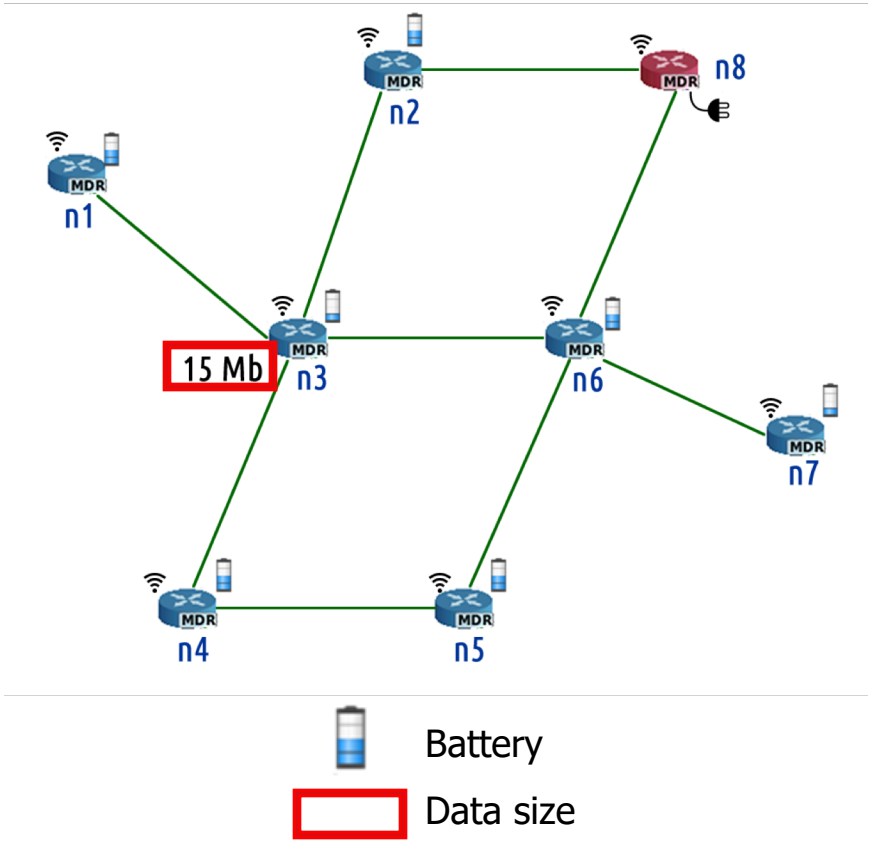
- We propose a fog computing technology to balance the energy consumption during data processing using a fog middleware.
- We first demonstrate that for a fog network with different kind of nodes and with no sink node, balancing the energy relies on a combinatorial optimization that is solved using graph theory.
- We propose a transformation of the transshipment graph problem to formulate an optimization problem that we solve with LP to find the optimal solution.
- We show the possibility of balancing and extending the battery life of the whole network based on cooperation between nodes without jeopardizing individual nodes' battery life.

Network Model



- In our model, we have a mesh network represented as a weighted graph G = (V, E, τ, λ), where each edge (u, v) $\in$ E has a transmission cost τ_{uv}, and a execution cost λ_{uv}.
- The network has a set of fog nodes and edge nodes that generate data to be processed locally or remotely.
- A task can be executed remotely in one or more nodes collaboratively if the data transmission cost is less than the execution cost.
- The data to be transmitted can pass through one or more hops.

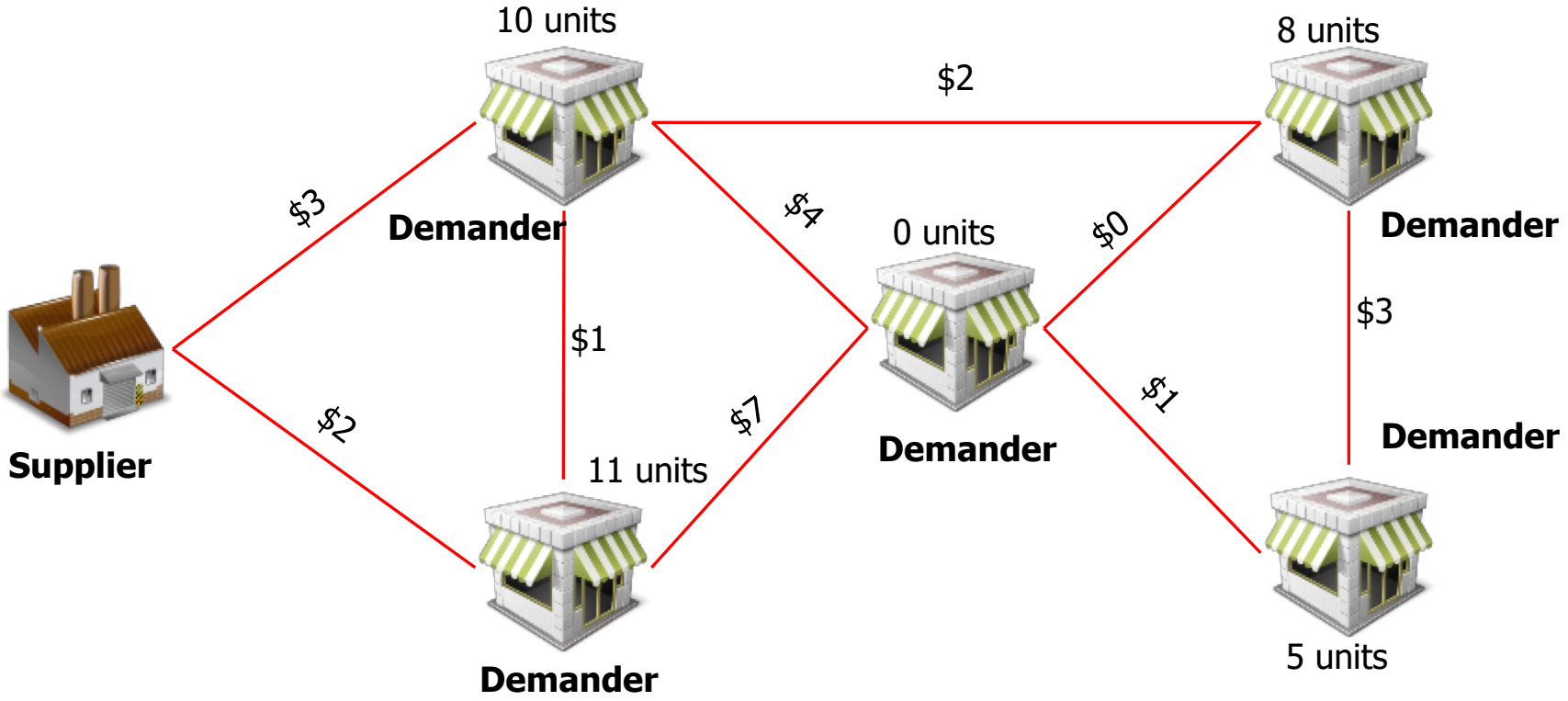
Network Model



- The batteries of some nodes can be running out faster than others.
- The data generated have different sizes.
- The execution cost can vary depending on the task that needs to be performed.
- We used a combined cost, that includes the transmission cost when the task is performed collaboratively and the execution cost in the nodes where the data are executed.

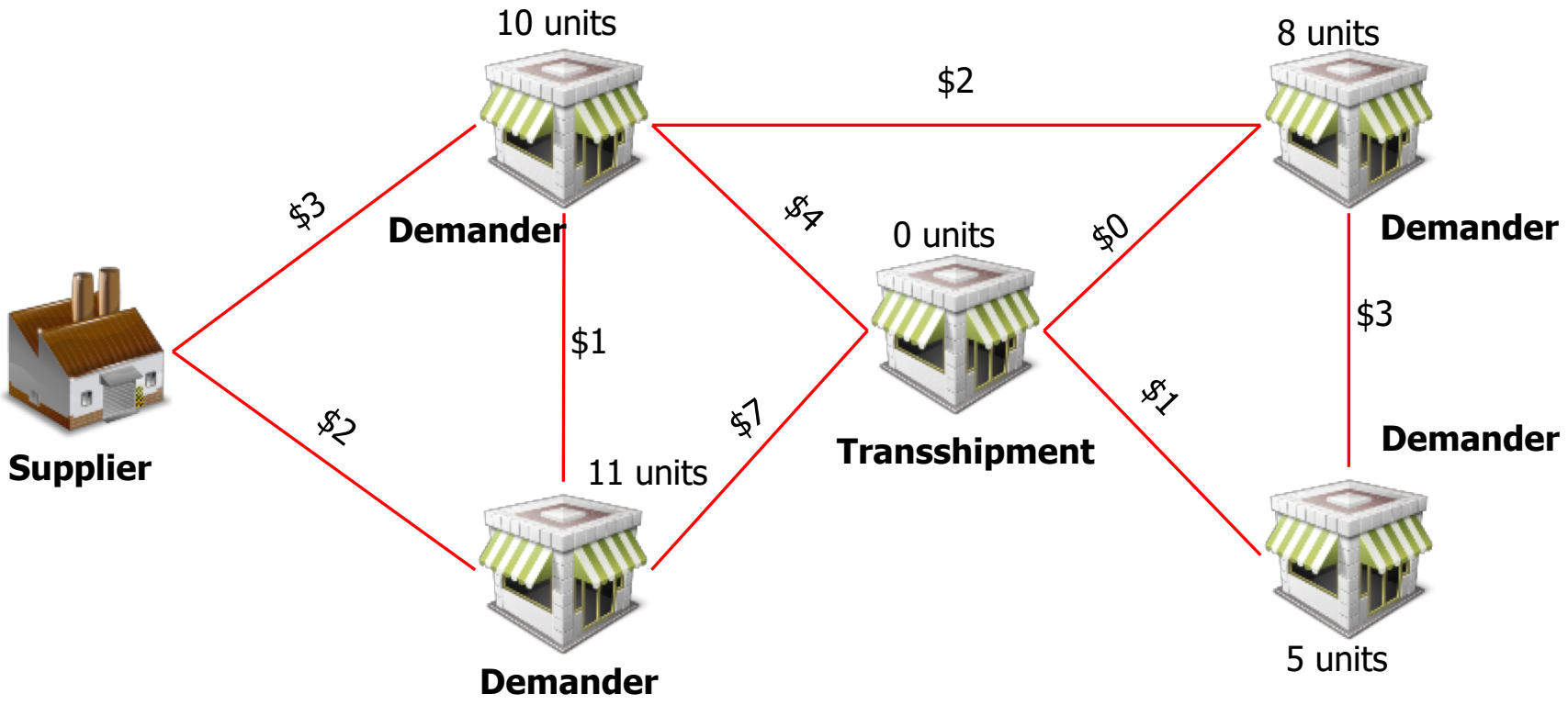
Graph Transformation - Transshipment

- The objective of the transshipment problem is to determine how many units should be shipped over each node so that all the demand requirements are met with the minimum transportation cost.
- The commodity can be transported to a particular destination through one or more intermediate or transshipment nodes.

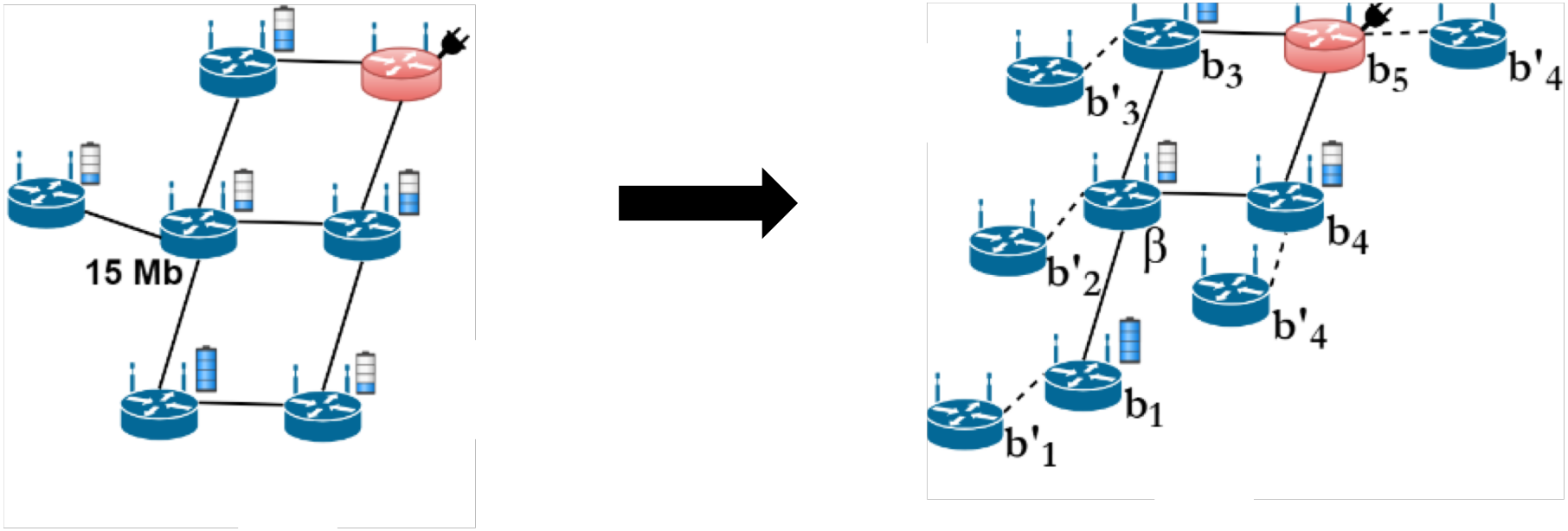


Graph Transformation - Transshipment

- We formulate the task distribution problem for balancing battery level as a transshipment problem from graph theory.

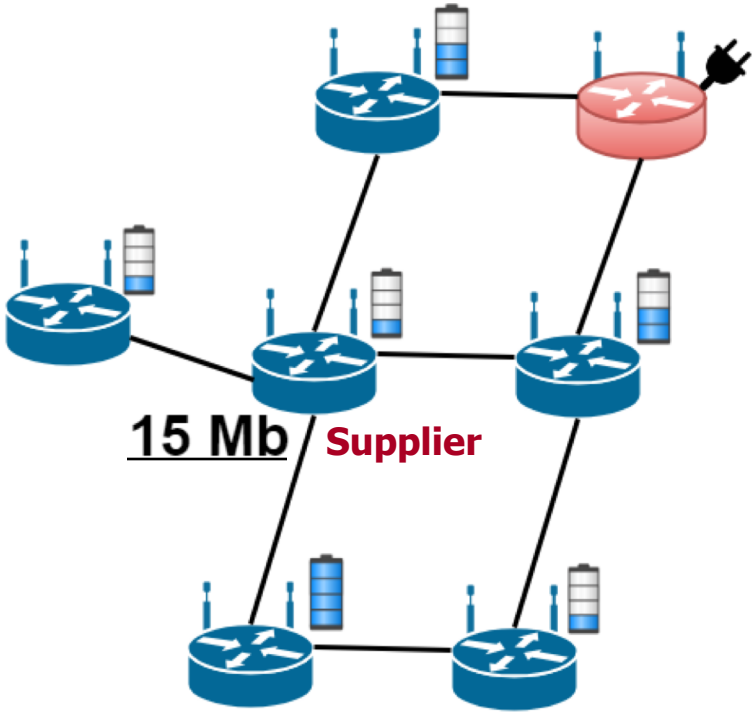


Graph Transformation - Goal



- Transform the mesh network to a transshipment graph network.

Graph Transformation - Definition



- The supplier is the node that needs to perform a specific task.
- The bytes of the data are the commodities.
- All nodes, including the supplier, are demander nodes.
- A node can or cannot get the whole amount of bytes that is demanded (it is calculated)

Graph Transformation - Definition

Model definition:

β → Amount of bytes per task

λ → Execution cost per byte

τ → Transmission cost per byte

δ → Residual capacity

b → Maximum number of bytes to be processed

Graph Transformation - Definition

Objective Function:

$$\text{minimize} \quad \sum_{(u,v)\in E} \lambda_{uv}\beta_{uv} + \sum_{(u,v)\in E} \tau_{uv}\beta uv \qquad (1)$$

Minimize the energy consumed from batteries

Subject to:

For supplier (source node):

$$\sum_{k\in K} \beta_{ku} - \sum_{l} \beta_{ul} \leqslant \beta \qquad (2)$$

Graph Transformation - Definition

Subject to:

For Demanders (target nodes):

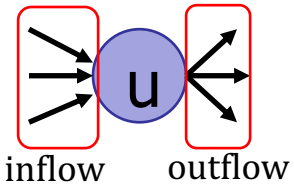
$$\sum_{k \in K} \beta_{ku} - \sum_{l \in L} \beta_{ul} \leqslant \mathrm{b}_u, \qquad (3)$$

$$\sum_{k \in K} \beta_{ku} - \sum_{l \in L} \beta_{ul} \leqslant \mathrm{b'}_u, \qquad (4)$$

where $\boldsymbol{K}$ and $\boldsymbol{L}$ are the subset of **inflow** and **outflow** nodes of $\boldsymbol{u}$ respectively.

$\boldsymbol{\beta_{ku}}$ represents the inflow (number of input bytes)

$\boldsymbol{\beta_{ul}}$ is the outflow (number of output bytes)



$$\boldsymbol{b_u} = \frac{\boldsymbol{\theta_u}}{\boldsymbol{\tau_u}} \qquad \boldsymbol{b'_u} = \frac{\boldsymbol{\theta_u}}{\boldsymbol{\lambda_u}}$$

θ represents the power balance or extra energy over the network average

$$\theta = \delta_u - ave\,(\delta_u) \forall\, u \in G$$

Graph Transformation – Transformation 1

The objectives of this transformation are:

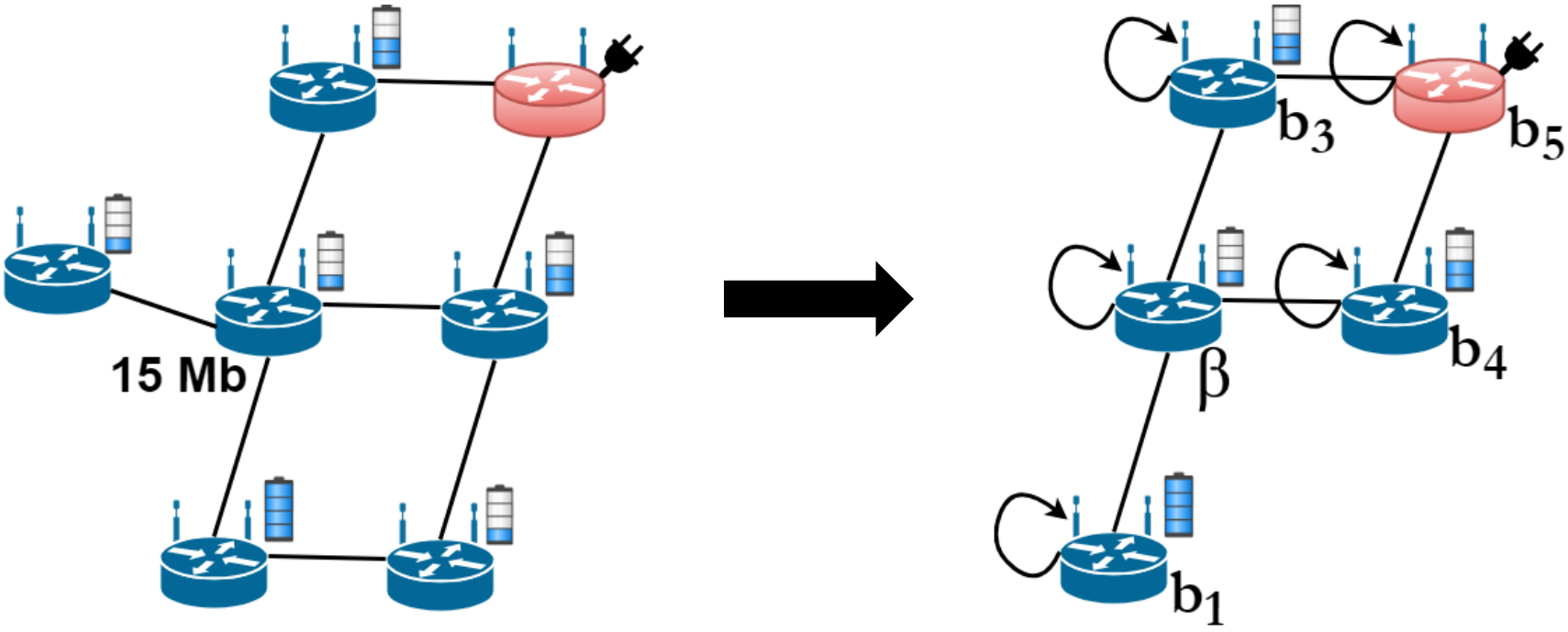
- Calculate the energy balance level of the network
- Eliminate unbalanced nodes
- Create the additional arc to satisfy the two costs
- Calculate the maximum amount of bytes that can be transmitted for each demander.

Graph Transformation – Transformation 1

Steps:

- Take any single-fog-node from the original graph and calculate its power balance.
- Then, the unbalanced demander nodes where $\theta < 0$ are pruned.
- The demander nodes who do not have connectivity with the supplier are removed.
- For the remaining nodes, the amount of bytes that can transmit with the energy of θ_u is calculated.
- The result is the maximum demand for each node $\boldsymbol{b_u} = \frac{\boldsymbol{\theta_u}}{\boldsymbol{\tau_u}}$
- The demand for edge-nodes is the whole task byte-size.
- Transmission cost is used in the arc connections.

Graph Transformation – Transformation 1



Original

After Transformation 1

Graph Transformation – Transformation 2

The objectives of this transformation are:

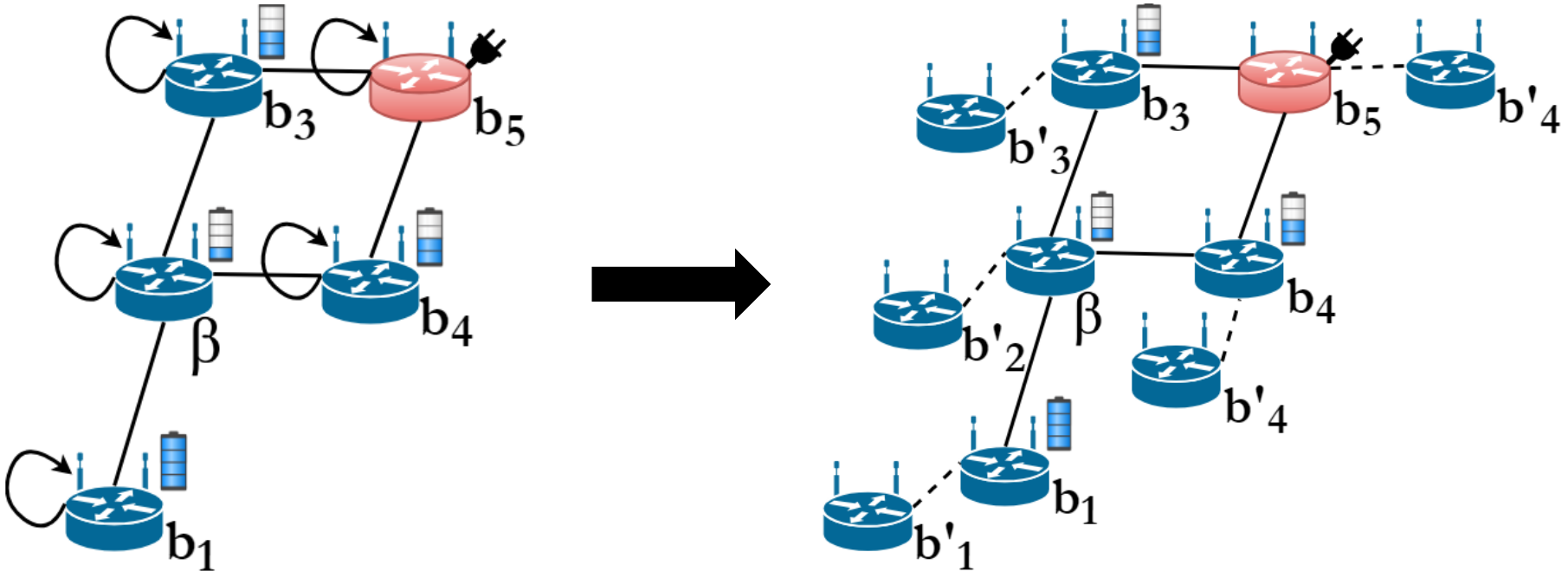
- Remove recursive loop
- Calculate the maximum amount of bytes that can be processed for each demander
- Establish transition and execution cost of edge nodes

Graph Transformation – Transformation 2

Steps:

- The transformation starts by using the graph from Transformation 1.
- For each node u another node labeled as u' is created.
- The loop on u is removed, and a new arc between u and u' is created, and the execution cost is used in the arc connections.
- The amount of bytes that can compute with the energy of θ_u is calculated.
- The result is the maximum demand for each node $\boldsymbol{b'_u} = \frac{\boldsymbol{\theta_u}}{\lambda_u}$
- For edges nodes, transition and execution cost are established as 0.

Graph Transformation – Transformation 2



After Transformation 1

After Transformation 2

Algorithm

- Every node broadcasts twice get the neighbor list one-hop neighbor list in order to obtain neighbors and their battery levels.
- The update stage that is executed every five minutes for updating the two-hop neighbor battery levels.
- The final stage is applied when a node needs to perform a task, and its battery is unbalanced with respect to its two-hop neighbors.
- Then, we apply the graph transformation to get the optimal solution for task distribution.

Algorithm 1 Optimal Data Task Distribution ODTD

procedure: *Initialization*

define Graph G adding node u
each node broadcast to get neighbor list
upon reception of ($energyLevel$) msg from v
for all $i \in nbrlist_u$ **do**
  add node i on G with $energyLevel_i$
  create arc between nodes u and i
end for
  broadcast u and $nbrlist_u$ with $energyLevels$ (covering all two-hop neighbors)
begin thread update energy levels
  for all $i \in nbrlist_v$ **do**
    if i $\notin$ G **then**
      add node i to G with $energyLevel_i$
      create arc between nodes u and i
    else
      update $energyLevel_i$ of node i
    end if
  end for
end thread update energy levels

procedure: *Optimal Distribution*

θ_u = calculate energy balance
if $energyLevel_u < \theta_u$ **then**
  G = apply transformation1 using G
  G = apply transformation2 using G
  $[X,Y]$ = solve transshipment using G
  for all $x \in X$ **do**
    send Y_x bites of data to x
  end for
  upon reception of $result$ from all X
end if

Analysis and Evaluation

Tests performed and parameters:

- We tested the optimization algorithm using an **emulated network** and a **real fog network.**
- The tasks to process in each node are related to pre-processed seismic data (**band-pass filter**).
- We first determined the costs of performing the band-pass filter over this kind of data, and the communication cost of transmitting the data.
- **Execution cost** of 2.31153e-9 energy units per sample.
- **Transmission cost** of the data in this scenario is 5.90944e-10 energy units per sample.
- We assume these costs as constant in our tests.

Analysis and Evaluation

1. Test using a **emulated network:**

- A network with 4 fog nodes only
- A network with 3 fog nodes and 1 edge node
- A network with 20 fog nodes
- A network with 18 fog nodes and 2 edge nodes.

- Two metrics were evaluated: the number of nodes alive at the final of the emulation, and the percentage time the nodes were active.

Analysis and Evaluation

Test using a **emulated network:**

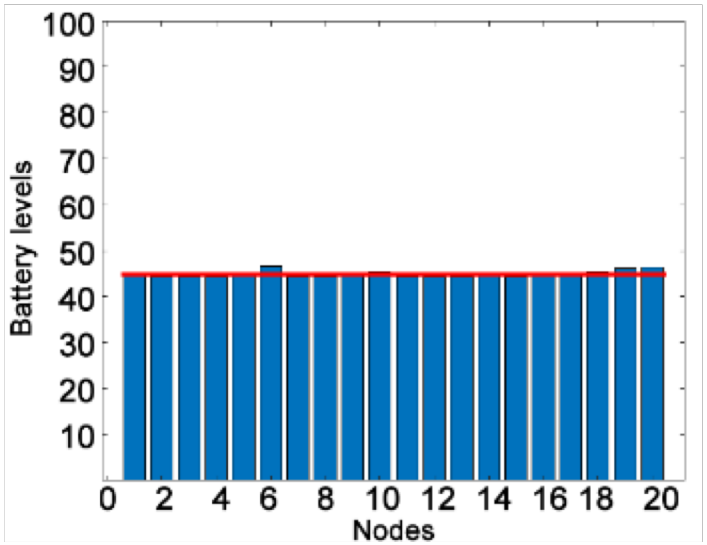
Table 1. Quantitative results of the emulation

Nodes	**4 Fogs**		**3 Fogs 1 Edge**	
Measurement	**Alive**	**Active**	**Alive**	**Active**
Without ODTD	2	62.78%	3	81.09%
ODTD	4	95.41%	4	98.27%
Nodes	**20 Fogs**		**18 Fogs 2 Edges**	
Measurement	**Alive**	**Active**	**Alive**	**Active**
Without ODTD	14	76.18%	15	83.61%
ODTD	20	97.33%	20	99.76%

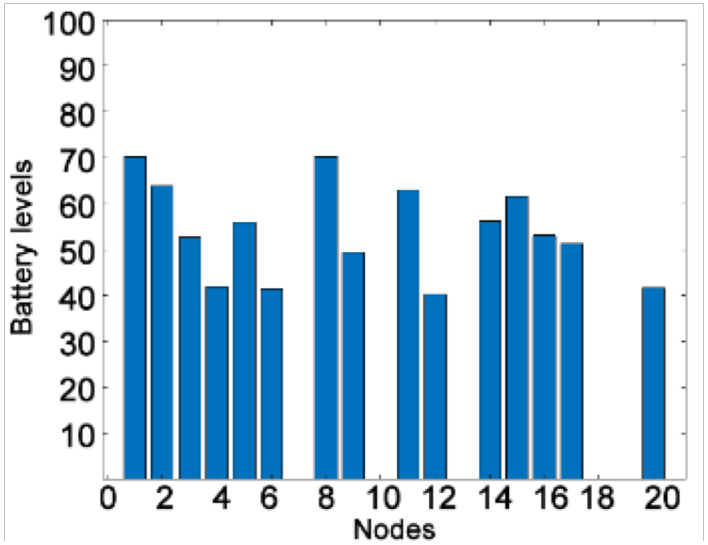
Analysis and Evaluation

Test using a **emulated network:**

- Final battery status in the scenario of 20 fog nodes



Using ODTD



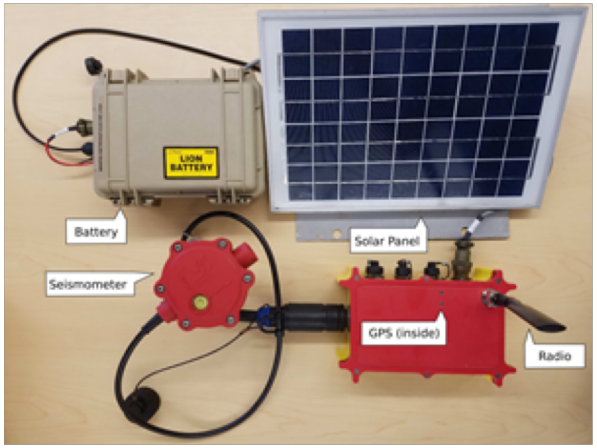
Without ODTD

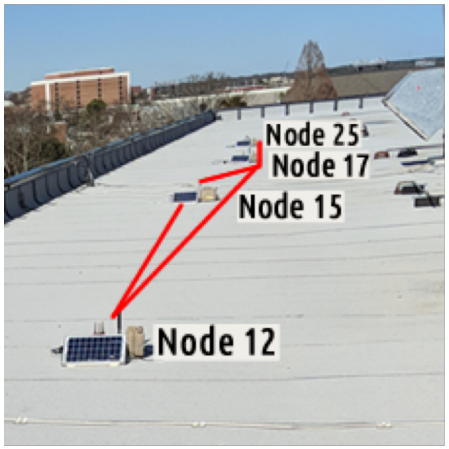
Results reveal that in addition to conserving the life of the nodes, the algorithm helps to synchronize the energy level of the batteries in the network.

Analysis and Evaluation

1. Test using a **real fog network:**

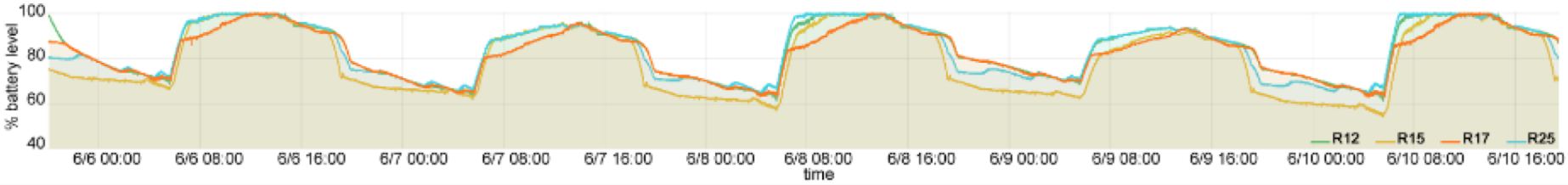
- Implemented ODTD algorithm in **Python** using **linprog** method to solve linear objective functions subject to inequality constraints.
- It was implemented in seismic units, and it was deployed on a roof.



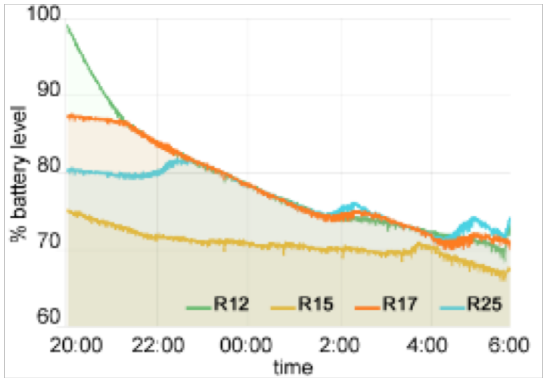


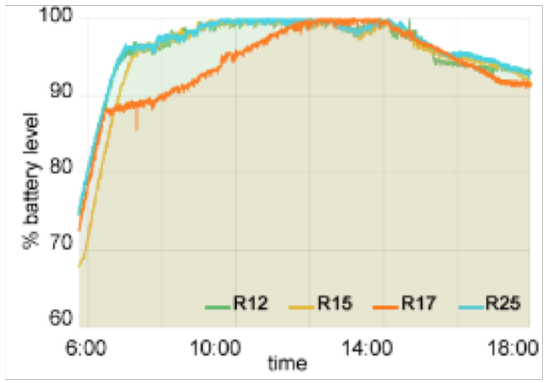
Analysis and Evaluation

1. Test using a **real fog network:**



Energy monitoring of the four units used in the deployment to validate the ODTD algorithm.





ODTD algorithm effect during the periods of unloading and loading of nodes batteries.

Conclusion

Remarks

- We formulated an in-network data task distribution as an optimization problem to extend the lifetime of nodes.
- We combined both transmission cost and execution cost into a single energy-efficient model to extend the battery energy and at the same time synchronize the battery energy level of the network.
- We designed ODTD, an efficient algorithm for optimal data task distribution that achieves the minimum energy consumption cost.
- An emulation was performed using Matlab to show its efficiency through different experimental scenarios.
- We implemented ODTD algorithm using Python and tested it on real seismic units for five days getting similar results and proving the algorithm feasibility.

References

[1] Ivan Stojmenovic and Sheng Wen, "The fog computing paradigm: Scenarios and security issues," in *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on*. IEEE, 2014, pp. 1–8.

[2] Fengyuan Ren, Jiao Zhang, Tao He, Chuang Lin, and Sajal K. Ren, "EBRP: energy-balanced routing protocol for data gathering in wireless sensor networks," *IEEE transactions on parallel and distributed systems*, vol. 22, no. 12, pp. 2108–2125, 2011.

[3] Jose Clemente, Maria Valero, Javad Mohammadpour, Xiangyang Li, and WenZhan Song, "Fog computing middleware for distributed cooperative data analytics," in *IEEE World Fog Congress 2017 (WFC 2017)*, 2017.

[4] Trust Tawanda, "A node merging approach to the transhipment problem," *International Journal of System Assurance Engineering and Management*, vol. 8, no. 1, pp. 370–378, 2017.

[5] Kai Lin, Min Chen, Sherali Zeadally, and Joel J. P. C. Rodrigues, "Balancing energy consumption with mobile agents in wireless sensor networks," *Future Generation Computer Systems*, vol. 28, no. 2, pp. 446–456, 2012.

[6] Seema Bandyopadhyay and Edward J. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*. IEEE, 2003, vol. 3, pp. 1713–1723.

[7] Fatma Bouabdallah, Nizar Bouabdallah, and Raouf Boutaba, "On balancing energy consumption in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 6, pp. 2909–2924, 2009.

[8] Yale T. Herer and Michal Tzur, "The dynamic transshipment problem," *Naval Research Logistics (NRL)*, vol. 48, no. 5, pp. 386–408, 2001.

[9] Liang Zhao, Wen-Zhan Song, Lei Shi, and Xiaojing Ye, "Decentralized seismic tomography computing in Cyber-Physical sensor systems," *Cyber-Physical Systems*, vol. 1, no. 2-4, pp. 91–112, 2015.

[10] Goutham Kamath, Lei Shi, Edmond Chow, and Wen-Zhan Song, "Distributed multigrid technique for seismic tomography in sensor networks," in *The 1st International Conference on Big Data Computing and Communication (BigCom)*, Shanxi, China., 2015, Springer, pp. 297–310.

[11] Lei Shi, Wen-Zhan Song, Fan Dong, and Goutham Kamath, "Sensor network for real-time in-situ seismic tomography," in *International Conference on Internet of Things and Big Data (IoTBD 2016)*, 2016.

[12] Maria Valero, Fangyu Li, Sili Wang, Fan-Chi Lin, and WenZhan Song, "Real-time cooperative analytics for ambient noise tomography in sensor networks," *IEEE Transactions on Signal and Information Processing over Networks*, 2018.

Thanks