Dynamic Load Balancing in Energy Packet Networks

A. Bušić [a,b] , J. Doncel [c] , J.M. Fourneau [d]

[a] Inria, Paris, France
[b] DI ENS, CNRS, PSL University, Paris, France
[c] University of the Basque Country, UPV/EHU, Leioa, Spain
[d] DAVID, UVSQ, Univ. Paris-Saclay, Versailles, France

Abstract

Energy Packet Networks (EPNs) model the interaction between renewable sources generating energy following a random process and communication devices that consume energy. This network is formed by cells and, in each cell, there is a queue that handles energy packets and another queue that handles data packets. We assume Poisson arrivals of energy packets and of data packets to all the cells and exponential service times. We consider an EPN model with a dynamic load balancing where a cell without data packets can poll other cells to migrate jobs. This migration can only take place when there is enough energy in both interacting cells, in which case a batch of data packets is transferred and the required energy is consumed (i.e. it disappears). We consider that data packet also consume energy to be routed to the next station. Our main result shows that the steady-state distribution of jobs in the queues admits a product form solution provided that a stable solution of a fixed point equation exists. We prove sufficient conditions for irreducibility. Under these conditions and when the fixed point equation has a solution, the Markov chain is ergodic. We also provide sufficient conditions for the existence of a solution of the fixed point equation. We then focus on layered networks and we study the polling rates that must be set to achieve a fair load balancing, i.e., such that, in the same layer, the load of the queues handling data packets is the same. Our numerical experiments illustrate that dynamic load balancing satisfies several interesting properties such as performance improvement or fair load balancing.

Keywords: Energy Packet Networks, Load Balancing, Fairness

1. Introduction

We are interested in analyzing a queueing network with dynamic load balancing. In these systems, jobs can migrate from one queue to another. This technique provides a good solution for distributed systems where the goal consists of achieving a fair workload distribution. Another advantage of dynamic load balancing is given when some queues are in the heavy-traffic regime, in which case migration of jobs can lead to reduce the load of these queues, which clearly improves the performance of the system. Dynamic load balancing techniques can be classified in two groups: receiver-initiated load balancing where an idle queue requests jobs to the rest of the queues, and sender-initiated load balancing where the queues transfer jobs when they are overloaded. The authors in [1] show that, in the heavy-traffic regime, the receiver-initiated load balancing performs better than the sender-initiated load balancing. This finding was also confirmed in a more recent work [2].

Receiver-initiated load balancing techniques are also known as work-stealing strategies in the literature and have been widely studied so far. For instance, in the context of parallel server systems, [3, 4] study work-stealing strategies using the mean field approach and [5] formulate the problem of optimal migration of low-priority jobs as a restless multiarmed bandit problem. In [6], a job is moved to an idle queue when the sender has more than a fixed threshold value of jobs. A closer related work to ours is [7] where the authors consider a queueing network with Poisson arrivals and exponential services times that operates under the received-initiated load balancing technique. More precisely, they consider that, when a queue gets empty, it polls another queue (which they call the sender queue) with an exponential time and brings a batch of jobs whose size is geometrically distributed and bounded by the number of jobs of the sender queue. Their main result shows that the steady-state distribution of packets in the queues has a product form expression. To the best of our knowledge, [7] is the first one to obtain a product-form solution for the steady-state distribution of networks with receiver-initiated load balancing.

The devices that form the modern communications systems consume energy. This energy comes often from renewable energy sources, which are clearly very volatile and introduce a high uncertainty about the amount of energy available in the network. As a result, several research works have been recently proposed queueing models that incorporate energy consumption of the queues and consider that the energy arrives to the system according to a random process. An example of these models is the Energy Packet Network (EPN) model, which has been introduced by Gelenbe and his colleagues in [8, 9, 10]. This model considers that energy is represented by packets of discrete units of energy, which we call Energy Packets (EPs), and the workload packets, which are called as Data Packets (DPs), are transmitted to the next station only when there is available energy. Thus, the network is divided in cells and, in each cell, there is a queue that handles EPs and another queue handling DPs. Most of the EPN models that have been explored in the literature are particular cases of G-networks [11, 12, 13], with a notable exception being [14] based on a Brownian motion model. Therefore, the known product-form results for G-networks extend to EPN models. There has been a recent interest of researchers in this field to analyze queueing models whose steady-state distribution of jobs has a product solution by including energy consumption using the EPN model, see [15] for a recent survey of EPN models.

In this article, we consider an EPN model where the DP queues start the transfer. This means that, when a DP ends service in a cell, it is sent to the EP queue of the same cell and it is transferred to the next cell if there are enough EPs, in which case the required number of EPs disappear. The main difference of our work with respect to the previously studied EPN models is that we consider that a load balancing of a batch DPs to moved from one cell to another as in [7]. However, in [7], the authors do not consider energy consumption. Therefore, in this work, we define how energy needs to be consumed for a successful load balancing. In fact, we consider that the load balancing is only successful when there are one EP in the polling cell (i.e., the cell that initiates the load balancing protocol) and a batch of EPs in the sender cell.

The main contributions of this work are summarized as follows:

- We show that the steady-state distribution of jobs in the queues of this model admits a product form solution provided that a stable solution of a fixed point equation exists. We give sufficient conditions under which the Markov chain is ergodic. We also provide sufficient conditions for the existence of the fixed point problem.

- We then focus on a layered network and we determine how the polling rates can be set so as to achieve load balancing in each layer, i.e., to equalize the load of all the DP queues of the same layer.

- Finally, we illustrate using numerical experiments that the load balancing technique we consider in this work has an important property that consists of improving the performance of the system by decreasing the load of the queues that are close to saturation. We also study numerically how the polling rate of layered networks can be computed.

The technical part of the paper is as follows. In Section 2, we present the model of an Energy Packet Network with load balancing as well as the network topology that we consider. Then in Section 3, we focus on a general network and prove the existence of a product form expression for the steady-state distribution of packets in the queues, analyze its ergodicity and provide sufficient conditions under which a solution of the fixed point equation exists. Section 4 is devoted to the analyze the parameters that ensure a fair load balancing in layered networks. Then in Section 5, we present some numerical examples to illustrate the main features of the model. In Section 6, we present the main conclusions of our work.

2. Model Description

2.1. The EPN model

We consider an EPN model with N cells. Cell i has two queues: one of them handles the DPs (DP queue i), whereas the other queue handles the EPs (EP queue i). We assume that DPs and EPs arrive to cell i following a Poisson process with rate λ_i and α_i, respectively, with $i = 1, \dots, N$. We also assume that a leakage of a EP at cell i occurs with exponential time with rate β_i, for $i = 1, \dots, N$.

We consider an EPN model in which DPs start the transfer and such that c_i EPs are required for a successful transfer to the next cell. More precisely, we assume that DPs are served at DP queue i following an exponential distribution with rate μ_i, for $i = 1, \dots, N$. When a DP ends service at cell i, it is sent to EP queue i and if there are less than c_i EPs, the DP is lost and EP queue i gets empty. Otherwise, (i.e., if a DP ends service at cell i where there are, at least, c_i EPs at cell i), the DP is routed to cell j with probability $P(i, j)$ and c_i EPs disappear from EP queue i. Hence, we have that for all i, $\sum_{j=1}^{N} P(i, j) = 1$. We also assume that a DP at cell i leaves the system after an exponential service time with rate δ_i and does not require energy.

2.2. Load Balancing with Energy Consumption

We consider in this EPN model a load balancing technique with energy consumption. It consists of a generalization to Energy Packet Networks of the model in [7] where the load balancing is based on polling. Thus, we assume, as in [7], that cell i initiates the polling protocol to get DPs from cell j with exponential time with rate $\gamma_{i,j}$. Furthermore, in this work, we assume that the load balancing consumes energy on both cells involved in the load balancing (note that the model of [7] does not consider energy consumption). We denote by $A_{i,j}^{(k)}$ the probability that k EPs at cell j are needed to carry out load balancing initiated from cell i (i.e. with probability $A_{i,j}^{(k)}$ k EPs are consumed when the load balancing to move DPs from cell j to cell i is successful), where $k = 0, 1, 2, \ldots$. We assume that for all i and j, $\sum_{k\geq 0} A_{i,j}^{(k)} = 1$, $A_{i,j}^{(0)} \neq 1$ and $\sum_k k \ A_{i,j}^{(k)} < \infty$.

In our model, the load balancing takes place if the following conditions are satisfied in order:

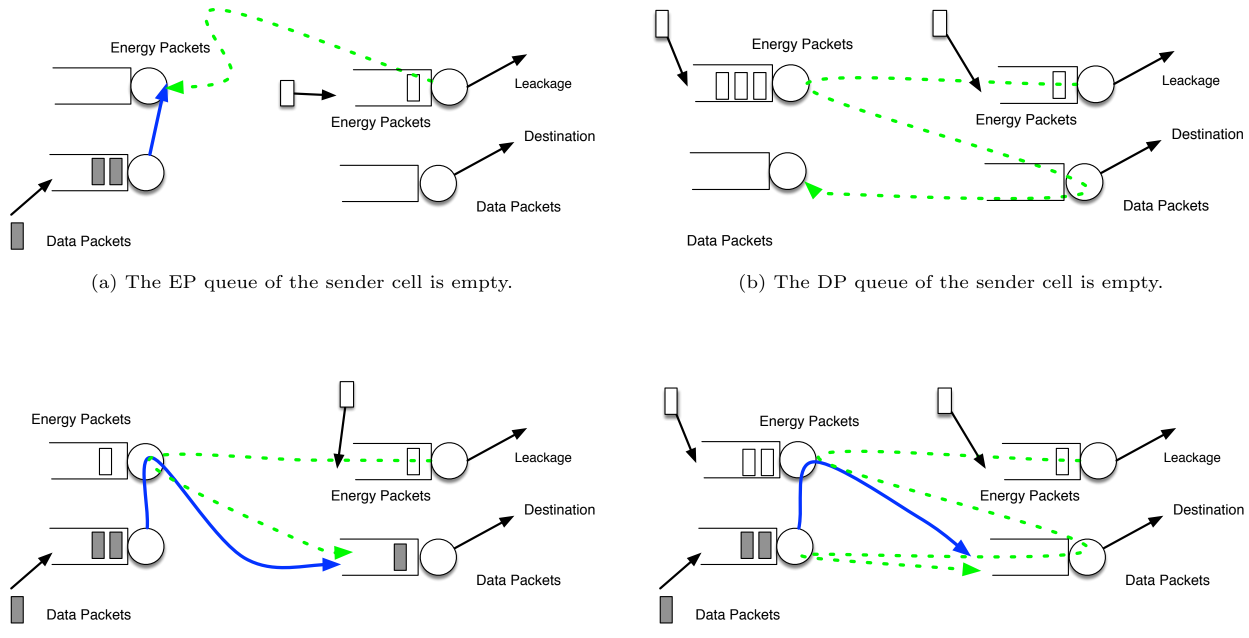
(COND1) There is at least one EP at cell i.

(COND2) There are at least k EPs at cell j.

(COND3) DP queue i is empty.

When the load balancing protocol gets activated, it first checks that (COND1) is satisfied. In the positive case, (COND2) establishes a second condition for the activation of the load balancing. In case (COND2) is not verified (i.e., if there is not enough energy at EP queue j), EP queue j gets empty and one EP at cell i disappears. When (COND1) and (COND2) are verified, (COND3) is the last condition to check, which consists of verifying that DP queue i is empty. If (COND3) is not satisfied, one EP of cell i and k EPs of cell j disappear. On the other hand, if (COND1), (COND2) and (COND3) are met, the load balancing operation initiated by cell i provokes a transfer of a batch of DPs from cell j to cell i, which requires 1 EP at cell i and k EPs at cell j. The number of DPs that are transferred from cell j to cell i follows a distribution denoted by $B_{i,j}$ which will be defined later.

In Figure 1, we represent four out of all the situations that can arise in a network with two cells. We depict the DPs as grey boxes, whereas the EPs as white boxes. Therefore, the queues containing white boxes are the EP queues and the queues containing grey boxes are the DP queues. We consider that the cell on the right polls for DPs to the cell on the left and also that the cell on the left sends DPs through routing. A successful load balancing and routing are only achieved for the case illustrated in (d). In fact, the load balancing occurs since (COND1), (COND2) and (COND3) are satisfied (there are EPs in the sender and receiver cells and the receiver DP queue is empty) and the routing is also successful since there are enough EPs in the left cell. In the rest of the cases, load balancing or the routing does not occur for different reasons. The caption of each figure presents the reason why the transfer (load balancing and/or routing) fails. In the case of (a), the routing cannot be done since there are not EPs in the left cell, whereas the load balancing is not successful since (COND1) is satisfied but (COND2) is not. In the case of (b), the DP queue of the left cell is empty and, therefore, there are no DPs to transfer due to routing and due to load balancing. This means that even if (COND1) and (COND2) are satisfied, we cannot transfer the DPs due to load balancing because the left cell is empty. In the case of (c), we observe that the routing is successful since there are DPs and EPs in the left cell, but the load balancing fails since the DP queue of the receiver is not empty, i.e., (COND3) is not satisfied.



(a) The EP queue of the sender cell is empty.

(b) The DP queue of the sender cell is empty.

(c) The DP queue of the receiver cell is not empty.

(d) A successful load balancing and routing.

Figure 1: The cell on the right polls for DPs the cell on the left. The solid blue arrow represents the interaction due to routing, whereas the green dotted arrow of load balancing.

Remark 1. *We would like to remark that the activation process for the load balancing is a new type of Domino synchronization which generalizes the ones studied in [16]. We would also emphasize that the model under study in this article differs with respect to [7] since in our model the load balancing requires energy. In other words, if there is not enough energy on both cells, the load balancing does not take place.*

Regarding the conditions that the load balancing protocol must verify, we notice that the ordering of (COND1) and (COND2) can be changed and the main result of this article, i.e., the existence of a product form expression of the steady-state distribution of the packets in the queues, is mantained. However, we have not been able to keep this property when (COND3) is verified first.

Let $B_{i,j}(X_j)$ be the batch distribution of DPs that migrate due to load balancing from cell j to cell i after a polling by cell i when there are X_j DPs at cell j. Let ρ_i (resp. ρ_j) be the DP load at cell i (resp. cell j). We now present the following assumption that will be useful to prove our main result.

Assumption 1. *$B_{i,j}(X_j)$ is a truncated geometric with rate $b_{i,j}$, where $b_{i,j} = \frac{\rho_i}{\rho_j}$ and $b_{i,j} < 1$. Thus,*

$$P(B_{i,j}(X_j) = k) = \begin{cases} (1 - b_{i,j})b_{i,j}^k, & \text{if } k < X_j, \\ b_{i,j}^k, & \text{if } k = X_j. \end{cases}$$

We present some properties of the distribution under consideration in Appendix A.

Remark 2. *The authors in [7] also consider this assumption on the batch distribution to prove that the steady-state distribution of packets in the queues admits a product form solution. From this assumption, it follows immediately that it is not possible to have both* $\gamma_{i,j} > 0$ *and* $\gamma_{j,i} > 0$ *(one cannot have simultaneously* $b_{i,j} < 1$ *and* $b_{j,i} < 1$ *since* $b_{i,j} = \frac{\rho_i}{\rho_j}$ *and* $b_{j,i} = \frac{\rho_j}{\rho_i}$ *). We also note that* $\rho_i/\rho_j < 1$ *implies that the transfer of DPs is carried out from queues with higher loads to queues to lower loads.*

We will be sometimes interested in the conditions such that the load of the queues that interact in the load balancing is equal. This will be called *fair* load balancing in this article.

We now present the set of transitions of this model. We denote by (X, Y) the state of the system, where X (resp. Y) is a vector in which the i-th element represents the number of DPs (resp. of EPs) in cell i. Let e_i be the vector with all zeros except for the i-th element which is a one.

- With rate λ_i, a DP arrives from outside to cell i, i.e., $(X, Y) \to (X + e_i, Y)$
- With rate α_i, an EP arrives to cell i, i.e., $(X, Y) \to (X, Y + e_i)$
- With rate δ_i, a DP of cell i leaves the system, i.e., when $X_i > 0$, $(X, Y) \to (X - e_i, Y)$
- With rate μ_i, a DP of cell i ends service and, when there are at least c_i EPs at cell i, it is sent to cell j with probability $P(i, j)$, and c_i EPs of cell i disappear, i.e., in this case we have the following transition $(X, Y) \to (X - e_i + e_j, Y - c_i e_i)$.

  When there $m < c_i$ EPs at cell i upon service of a DP at cell i, the DP disappears and the EP queue of cell i gets empty, i.e., in this case we have the following transition $(X, Y) \to (X - e_i, Y - mc_i)$.
- With rate β_i, a EP of cell i is lost, i.e., when $Y_i > 0$, $(X, Y) \to (X, Y - e_i)$
- With rate $\gamma_{i,j}$ the load balancing protocol to migrate jobs from cell j to cell i is activated. For this case, we have that, with probability $A_{i,j}^{(k)}$, k EPs in cell i are required for a successful transfer and, also in case of a successful transfer, with probability $B_{i,j}^{(m)}(X_j)$, we migrate m DPs from cell j to cell i. Hence, the following transitions are given in the load balancing protocol:
  - When the EP queue of cell i is not empty and there are l EPs in cell j, with $l < k$, (COND2) is not verified and the EP queue of cell j gets empty and one EP in cell i is lost, i.e., $(X, Y) \to (X, Y - e_i - le_j)$
  - When the EP queue of cell i is not empty and there are, at least, k EPs in cell j, but the DP queue of cell i is not empty, (COND3) is not verified, in which case one EP of cell i and k EPs of cell j are lost, i.e., $(X, Y) \to (X, Y - e_i - ke_j)$

– When the EP queue of cell i is not empty, there are, at least, k EPs in cell j and the DP queue of cell i is empty, i.e., the load balancing is successful, we have that one EP of cell i and k EPs of cell j are lost and m DPs are moved from cell j to cell i, i.e., $(X, Y) \to (X + me_i - me_j, Y - e_i - ke_j)$

2.3. Network topology

Let us define the following directed graphs (digraphs): $\mathcal{R} = (V, E)$ and $\mathcal{G} = (V, L)$ where V is the set of cells, i.e. $V = \{1, \ldots, N\}$, E is the set of arcs which represent the possible movements of DPs after getting service in the corresponding DP queue and L is the set of arcs which represent the movements of DPs due to load balancing. We also define $\mathcal{H} = (V, E \cup L)$. In Figure 2, we plot a EPN network with load balancing with 4 cells and in Figure 3 its corresponding digraphs $\mathcal{R}$ and $\mathcal{G}$.

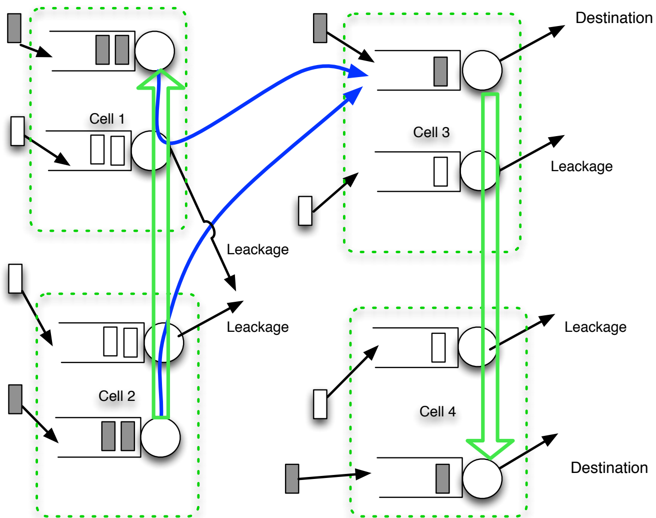


Figure 2: An example of an EPN model with load balancing formed by 4 cells. EPs are depicted as white boxes and DPs as grey boxes. The cells (one DP queue and the associated EP queue) are green boxes. The blue arrows represent the arcs in E (i.e., the movement of DPs after getting service), whereas the green arrows represent the arcs in L (i.e., the load balancing movements). Cell 1 is polling Cell 2 (i.e. $\gamma_{1,2} > 0$) while Cell 4 is polling Cell 3 ($\gamma_{4,3} > 0$).

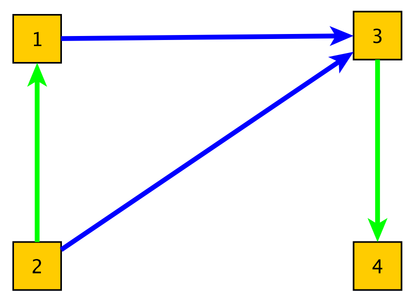


Figure 3: The digraphs associated to the EPN model of Figure 2. Blue arcs represent the edges of digraph $\mathcal{R}$ and green arcs the edges of digraph $\mathcal{G}$.

From what we said in Remark 2, the assumption on the batch distribution implies that we cannot have an arc (i, j) and an arc (j, i) in $\mathcal{G}$, i.e., $\mathcal{G}$ does not have directed cycles of length 2. One can generalize this property to a directed cycle with any number of nodes.

Proposition 1. *There is no directed cycles in the load balancing graph. Equivalently $\mathcal{G}$ is a directed acyclic graph.*

Proof. Consider an arbitrary directed cycle $i_1, i_2, .., i_k, i_1$ in graph $\mathcal{G}$. Assuming that one can balance the load between nodes i_j and i_{j+1}, one must have $\rho_{i_j} < \rho_{i_{j+1}}$ for all j. Combining these relations for all indices

i_j, we get:

$$\rho_{i_1} < \rho_{i_2} < \ldots < \rho_{i_k} < \rho_{i_1},$$

which is a clear contradiction. Therefore such a directed cycle does not exist. $\square$

It is important to notice that the above result implies that, in this model, the energy can not be wasted with an inefficient load balancing. Indeed, a directed cycle in $\mathcal{G}$ means that a DP may join the same queue after a sequence of load balancing operations. As these operations consume energy and do not process the workload, from the practical point of view it is important to avoid such loops.

Definition 1. *Cell i is a sink if at least one of the two conditions hold:*

- $\delta_i > 0$,
- $\mu_i > 0$ *and* $\beta_i > 0$.

Intuitively, a sink is a cell where the data packets can leave the system either because they arrive at their destination (i.e. $\delta_i > 0$) or because they disappear in a failed routing due to an empty energy buffer provoked by the leakage.

Definition 2. *An EPN model is open if the following conditions hold:*

1. $\alpha_i > 0$ *for all* i,
2. *for all DP queue j, there exist a DP queue i such that $\lambda_i > 0$ and there exist a path from i to j in graph $\mathcal{H}$,*
3. *for all DP queue j, there exist a cell i which is a sink and such that there exist a path from j to i in graph $\mathcal{H}$,*
4. *for all EP queue i, the condition $\beta_i + \sum_j \gamma_{i,j} > 0$ must hold.*

Intuitively, the first two conditions imply that every cell receives EPs and DPs while the third one states that DPs may leave the network cell due to service or load balancing. Finally the fourth condition states that the EPs may leave the system even if there is no data packet in the cell. Remark that the second and the third condition do not imply that the network is connected. However we assume in the following and without loss of generality that directed graph $\mathcal{H}$ is connected.

Lemma 1. *If the network is open, then there exists a directed rooted forest which spans graph $\mathcal{H}$ whose roots are such that the arrival rate of fresh data packets are positive: (i.e. $\lambda_i > 0$ if i is the root of a tree) and such all the edges point from the roots (we consider out-trees here). Recall that such a structure is called an out-forest. Such an out-forest induces a labeling of the cells such that if there is a path from cell carrying label i to cell labeled j, then we must have $i < j$. Let $I()$ be these labels.*

Similarly, one can obtain another labeling of the cells called $J()$ based on the following rooted in-forest. The roots are the sinks of the network and we consider a spanning out-forest where the nodes point toward the roots.

Proof. We first consider all the cells i such that $\lambda_i > 0$. Assume that we have k such cells. These nodes receive labels from 1 to k. We then build the out-trees rooted in these nodes by a Breadth First Search (BFS) algorithm among the remaining nodes of $\mathcal{H}$. Such an out-forest induces the required ordering by using a increasing sequence of numbers during the BFS visits.

The proof for labeling $J()$ is similar. We give the smallest number to the sinks of the network. Then we proceed by a breadth first search visits using the arcs in the opposite direction. □

Intuitively, labels $I()$ are used to describe the input of DP while labels $J()$ model how the DP leave the network.

In Table 1, we summarize the main notation of this article:

Notation	Description
λ_i	Arrival rate of DP queue i
α_i	Arrival rate of EP queue i
β_i	Leakage rate of EP queue i
μ_i	Service rate of DPs at cell i
c_i	The number of EPs required for a routing of a DP at cell i
δ_i	Departure rate of DPs at cell i
$P(i,j)$	Probability to route a DP from cell i to cell j
$\gamma_{i,j}$	Polling rate of cell i to cell j
$A_{i,j}^{(k)}$	Probability that the migration of jobs from cell j to cell i consumes k EPs
$B_{i,j}$	Batch size of migrating jobs from cell j to cell i
ρ_i	Load of DPs at cell i
ω_i	Load of EPs at cell i
$\mathcal{R}$	Routing graph
$\mathcal{L}$	Load balancing graph
$\mathcal{H}$	Routing or load balancing graph

Table 1: The main notation of this article.

3. Analysis in a Network with an Arbitrary Open Topology

We recall that X is the vector of the number of DPs in the DP queues and Y the vector of the number of EPs in the EP queues. Under the aforementioned assumptions, $(X,Y)_t = (X_1, ...X_N, Y_1, .., Y_N)_t$ is clearly a Markov chain. We first consider an arbitrary EPN model and prove that there exists an invariant measure of the packets in the queues that has a product form distribution provided that there exists a solution to a fixed-point equation such that $\rho_i < 1$ and $\omega_i < 1$ for all i. Under these conditions, when the EPN model is open, the Markov chain is ergodic. Finally, we provide sufficient conditions for the existence of a solution to the fixed point equation.

Consider the flow equation of the EPs:

$$\omega_i = \frac{\alpha_i}{\beta_i + \mu_i \rho_i \sum_{m=0}^{c_i-1} \omega_i^m + \sum_j \gamma_{i,j} + \sum_j \sum_{k\geq 1} \sum_{l=0}^{k-1} A_{j,i}^{(k)} \omega_j (\omega_i)^l \gamma_{j,i}}, \tag{FLOW-EP}$$

and the flow equation of the DPs:

$$\rho_i = \frac{\lambda_i + \sum_j \mu_j \rho_j \omega_j^{c_j} P(j,i) + \sum_j \sum_{k\geq 0} (A_{i,j}^{(k)} \gamma_{i,j} \omega_i \omega_j^k \rho_i - A_{j,i}^{(k)} \gamma_{j,i} \omega_j \omega_i^k \rho_j)}{\mu_i + \delta_i}. \tag{FLOW-DP}$$

In the following result, we prove that the steady-state distribution of packets in the queues admits a product form expression if there exists a solution of the fixed-point equations (FLOW-DP) and (FLOW-EP) such that $\rho_i < 1$ and $\omega_i < 1$. The proof is given in Appendix B.

Theorem 1. *Consider that the flow equation* (FLOW-EP) *and* (FLOW-DP) *have a solution such that for all $i \in V$ $\rho_i < 1$ and $\omega_i < 1$. Under Assumption 1, the following expression is an invariant probability measure*

$$\pi(X,Y) = \left(\prod_{i=1}^{N} (1-\rho_i)\rho_i^{X_i}\right)\left(\prod_{i=1}^{N} (1-\omega_i)\omega_i^{Y_i}\right). \tag{1}$$

Let us now present the following result about ergodicity which follows directly from Brémaud [17].

Theorem 2 ([17])**.** *Assume that the Markov chain is irreducible and that a solution of the flow equation exists such that $\rho_i < 1$ and $\omega_i < 1$ for all cell i, then $\pi(X,Y)$ is a distribution of probability and the Markov chain is ergodic.*

We now provide the following result regarding irreducibility.

Lemma 2. *If the network is open, then the Markov chain is irreducible.*

Proof. The proof has two parts: first we establish that there exists a sequence of transition with positive probability to lead from state $(\vec{0},\vec{0})$ to any state (X,Y). Then we prove that there also exists a sequence of transitions in the chain to connect (X,Y) to $(\vec{0},\vec{0})$. The proof is postponed to Appendix C.

Remark 3. *To emphasize that the above condition is sufficient, we provide in Appendix D an example of an irreducible network which is not open.*

As a consequence of the above two results, the next one follows directly.

Corollary 1. *Assume that the network is open and that a solution of the flow equations* (FLOW-DP) *and* (FLOW-EP) *exists such that $\rho_i < 1$ and $\omega_i < 1$ for all cell i, then $\pi(X,Y)$ is a distribution of probability and the Markov chain is ergodic.*

However, finding necessary and sufficient conditions for the existence of these solutions is still an open problem due to the multiple interactions between the queues. Indeed, when a successful load balancing operation takes place, 4 queues are modified (2 DP queues and 2 EP queues). This precludes the existence of a loop free sequence of numerical computation of the flow equation which is a possible way to prove existence of a fixed point solution (for instance for G-networks). In the following, we focus on a sufficient condition for the existence of a solution of (FLOW-EP) and (FLOW-DP) such that $\rho_i < 1$ and $\omega_i < 1$ for all i.

Let $\omega = (\omega_1, \ldots, \omega_N)$ and $\rho = (\rho_1, \ldots, \rho_N)$. We consider the following system of equations:

$$\omega_i = F_i(\omega, \rho) = \frac{\alpha_i}{\beta_i + \mu_i \rho_i \sum_{m=0}^{c_i - 1} \omega_i^m + \sum_j \gamma_{i,j} + \sum_j \sum_{k \geq 1} \sum_{l=0}^{k-1} A_{j,i}^{(k)} \omega_j (\omega_i)^l \gamma_{j,i}},$$

and

$$\rho_i = G_i(\omega, \rho) = \frac{\lambda_i + \sum_j \mu_j \rho_j \omega_j^{c_j} P(j,i) + \sum_j \sum_{k \geq 0} (A_{i,j}^{(k)} \gamma_{i,j} \omega_i \omega_j^k \rho_i - A_{j,i}^{(k)} \gamma_{j,i} \omega_j \omega_i^k \rho_j)}{\mu_i + \delta_i}.$$

We aim to find sufficient conditions for such a system to have a solution in R_+^{2N}. We first define the notion of hyper-stability of an EPN model.

Definition 3. *[Hyper-stability] A network is hyper-stable if the following conditions hold:*

(HYP1) $0 < \alpha_i < \beta_i + \sum_j \gamma_{i,j}$ *for all* i,

(HYP2) $\lambda_i > \sum_j \gamma_{j,i}$ *for all* i,

(HYP3) The Jackson network with arrival rates $(\lambda_i + \sum_j \gamma_{i,j}) \frac{\mu_i}{\mu_i + \delta_i}$, *service rate* μ_i *and routing matrix defined as* $\tilde{P}(j,i) = \frac{\mu_i}{\mu_i + \delta_i} P(j,i)$ *is positive recurrent and the solution* ρ_i^* *of this Jackson network is such that* $\rho_i^* < 1$ *for all* i,

(HYP4) $\mu_i + \delta_i > 0$ *for all* i.

Intuitively, (HYP1) implies that every EP queue receives and loses energy and it means that the EP queues are stable even when energy is not consumed for the routing and the customer movements (service and polling). (HYP2) states that, for each queue, the arrival rate of fresh customers must be larger the rate with which it is polled. And (HYP3) means that the network built taking into account the arrivals provoked by the load balancing (but not the departures) is stable. Conditions (HYP4) implies that every DP queue has a service capacity. They are slightly more restrictive than the open topology hypothesis. Due to these three last assumptions, the functions $F_i()$ and $G_i()$ are continuous on R_+^{2N}. We state that if the conditions of Definition 3 hold, then there exists a fixed point solution of (FLOW-EP) and (FLOW-DP). The proof is presented in Appendix E.

Theorem 3. *If the network is hyper-stable, then there exists a solution for the flow equation such that for all* $i \in V$ $\rho_i < 1$ *and* $\omega_i < 1$.

4. Analysis in an Arbitrary Open Layered Network

According to Remark 2, we have that load balancing transfers DPs from cells with high load to cells with low load. This clearly contributes to avoid that the load of DPs at some cells is close to one, which clearly improves the performance of the system. A possible way to achieve this goal is to find the polling rates such that a fair load balancing is given, i.e., the load of DPs of all the cells is equal. For this purpose, we consider an open layered network. Intuitively, an open layered network is divided in blocks or layers and each block is formed by cells with the same routing probabilities of movement after being served. Furthermore, load balancing is only allowed between cells of the same block, i.e., there is no arc in L between nodes of different blocks.

In this section, we provide a methodology such that, in each block, a fair load balancing can be achieved, i.e., the load of DPs of each of the same block is equal. In Section 4.1, we show how load balancing can be obtained in the first block. Then, in Section 4.2, we show how we can obtain by induction load balancing in the rest of the blocks.

Let us first define the networks under analysis in this section.

Definition 4. *A network is layered if graphs $\mathcal{R}$ and $\mathcal{G}$ are built as follows:*

1. *The connected components of the undirected version of $\mathcal{G}$ are denoted by K_1,...,K_{cc}. The subsets K_i are the layers of the EPN model and cc is the number of layers.*
2. *If x and y are in the same subset K_i, then there is no path from x to y in graph $\mathcal{R}$.*
3. *If node x in K_i has a positive routing probability in $\mathcal{R}$ to reach node y in K_j, then all the nodes of K_i have a positive probability to reach a node (not necessarily y) in K_j.*
4. *Routing graph $\mathcal{R}$ is feed forward (or a DAG using a graph terminology).*
5. *$\mu_i + \delta_i > 0$ for all queue i.*

Without loss of generality, we assume that the routing is only possible to join a subset K_i with a higher index.

In Figure 2, we represent a network with two layers; one layer is formed by cell 1 and cell 2, whereas the other is formed by cell 3 and cell 4. We observe that load balancing (which is represented by the red arrow) is given only between cell 1 and cell 2 and between cell 3 and cell 4, i.e., between nodes of the same layer. We also note that all the jobs that are routed after getting service at cell 1 and cell 2 to cell 3. In Figure 4 we represent another example of an open layered network.

We want to achieve a fair load balancing within each layer. Inside the layer K_i, all the DP queues will have the same load ρ_{K_i}, but the load may be distinct for all the layers, i.e., $\rho_{K_i} \neq \rho_{K_j}$, for $i \neq j$. We now present the following assumption we make in this section:

Assumption 2. *We assume that $A_{i,j}^{(k)} = 1$ for $k = 1$, i.e., one EP is required in cell j to carry out a load balancing that transfers DPs from cell j to cell i.*

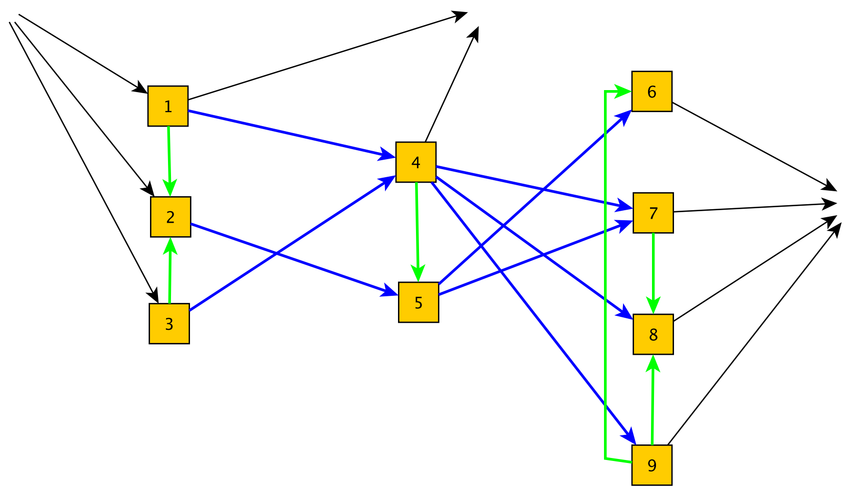


Figure 4: A layered network with 3 layers. We only represent the DP queues. The green arrows represent the load balancing, while the blue ones model the routing. The arrivals and departures are depicted by black arcs.

We would like to remark that the analysis of this section does not require the above assumption, but it is made to simplify the analytical and numerical analysis of this section.

Based on the topological properties of this network, we analyze the system in the topological order of the subsets K_i. Such an ordering exists due to the assumptions on a layered network.

4.1. Analyzing K_1

We now focus on the fair load balancing of K_1, i.e., of the first block of the layered network. Our approach is divided in three steps. First, in Section 4.1.1, we find the cells that must transfer jobs and which are the cells that receive jobs. Second, in Section 4.1.2, we present how to obtain the excess value of the sender cells and the deficit value of the receiver cells. Finally, in Section 4.1.1, we describe how to obtain the values of the polling rate to achieve load balancing using the excess and deficit values.

Let us consider (FLOW-DP) and, doing the summation for all the nodes i in K_1, it results:

$$\sum_{i\in K_1} \rho_i(\mu_i+\delta_i) = \sum_{i\in K_1} \lambda_i + \sum_{i\in K_1}\sum_{j} \mu_j\rho_j\omega_j^{c_j} P(j,i) + \sum_{i\in K_1}\sum_{j} (\gamma_{i,j}\omega_i\omega_j\rho_i - \gamma_{j,i}\omega_j\omega_i\rho_j) \tag{2}$$

But, as i is a node of K_1, it does not receive any DP from the network, i.e., $P(j,i)=0$ for all j. And since in an open layered network the load balancing only takes place between nodes of K_1, the above equation is simplified as follows:

$$\sum_{i\in K_1} \rho_i(\mu_i+\delta_i) = \sum_{i\in K_1} \lambda_i + \sum_{i\in K_1}\sum_{j\in K_1} (\gamma_{i,j}\omega_i\omega_j\rho_i - \gamma_{j,i}\omega_j\omega_i\rho_j) \tag{3}$$

We now remark that $\sum_{i\in K_1}\sum_{j\in K_1}(\gamma_{i,j}\omega_i\omega_j\rho_i - \gamma_{j,i}\omega_j\omega_i\rho_j) = 0$. Therefore, we get:

$$\sum_{i\in K_1} \rho_i(\mu_i+\delta_i) = \sum_{i\in K_1} \lambda_i \tag{4}$$

Thus, when the fair load balancing is achieved, we have that $\rho_i = \rho_{K_i}$ for all $i \in K_1$, it follows that:

$$\rho_{K_1} = \frac{\sum_{i \in K_1} \lambda_i}{\sum_{i \in K_1} (\delta_i + \mu_i)}.$$

The question is still to find the load balancing rates $\gamma_{i,j}$ to obtain such a fair balance. We proceed in two steps: first we find the non zero entries of the load balancing matrix to obtain a fair load balancing and then we derive algorithmically the rates $\gamma_{i,j}$.

4.1.1. Finding the interacting cells

Note that several choices are possible leading to many distinct ways to obtain a fair load balancing in K_1. We propose a simple method which has a low complexity. It may be possible to derive other methods to optimize other objectives (for instance energy consumption).

Let $\rho_i^{(I)}$ be the load of DPs at cell i when the polling rates are all equal to 0, i.e., $\rho_i^{(I)} = \frac{\lambda_i}{\mu_i + \delta_i}$. We partition the set of cells of K_1 into three subsets:

1. cells i such that $\rho_i^{(I)} < \rho_{K_1}$ (subset $V_{K_1}^-$),
2. cells i such that $\rho_i^{(I)} > \rho_{K_1}$ (subset $V_{K_1}^+$),
3. cells i such that $\rho_i^{(I)} = \rho_{K_1}$ (subset $V_{K_1}^=$).

Note that the set $V_{K_1}^-$ is formed by the nodes with deficit values and the set $V_{K_1}^+$ by the nodes of excess values. Therefore, we need to transfer load from the nodes of V^+ to the nodes of V^-. Moreover, the cells of $V_{K_1}^=$ are already balanced and, therefore, we do not use them to achieve the load balancing, i.e., we have that $\gamma_{i,j} = 0$ if $i \in V_{K_1}^=$ or $j \in V_{K_1}^=$.

From (FLOW-DP), it follows that, after fair load balancing, for all $i \in K_1$:

$$\rho_{K_1} = \frac{\lambda_i + \rho_{K_1} \sum_j (\gamma_{i,j}\omega_i\omega_j - \gamma_{j,i}\omega_j\omega_i)}{\delta_i + \mu_i}. \tag{5}$$

From the partition we have defined above, it follows that the balancing of DPs is achieved by poling cells in $V_{K_1}^+$ by cells in $V_{K_1}^-$.Remember that $\gamma_{i,j} > 0$ means that cell i polls cell j to receive DPs. Therefore $\gamma_{i,j} = 0$, if $i \in V_{K_1}^+$ or $j \in V_{K_1}^-$. Note also that the graph is bipartite. As a result, from (5), it follows that for a cell $i \in V_{K_1}^-$:

$$\rho_{K_1} = \frac{\lambda_i + \rho_{K_1} \sum_{j \in V_{K_1}^+} \gamma_{i,j}\omega_i\omega_j}{\delta_i + \mu_i} \iff \left(\rho_{K_1} - \rho_i^{(I)}\right)(\mu_i + \delta_i) = \rho_{K_1} \sum_{j \in V_{K_1}^+} \gamma_{i,j}\omega_i\omega_j,$$

whereas for a cell j in $V_{K_1}^+$:

$$\rho_{K_1} = \frac{\lambda_j - \rho_{K_1} \sum_{i \in V_{K_1}^-} \gamma_{i,j}\omega_i\omega_j}{\delta_j + \mu_j} \iff \left(\rho_{K_1} - \rho_i^{(I)}\right)(\mu_j + \delta_j) = -\rho_{K_1} \sum_{i \in V_{K_1}^-} \gamma_{i,j}\omega_i\omega_j.$$

We now aim to determine the values of $\gamma_{i,j}$ for all $i \in V_{K_1}^-$ and all $j \in V_{K_1}^+$ to achieve a fair load balancing. First, we will find the value of the excess and deficit values of the interacting cells, i.e., $\rho_{K_1}\gamma_{i,j}\omega_i\omega_j$ for each $i \in V_{K_1}^-$ and each $j \in V_{K_1}^+$ and then we will focus on the values of ω_i and ω_j to compute the polling rate to achieve load balancing.

4.1.2. Determining the excess and deficit values for each $i \in V_{K_1}^{+}$ and each $j \in V_{K_1}^{-}$

We aim to analyze how the excess values are balanced to the nodes with deficit values. Hence, our goal is to get $\rho_{K_1}\gamma_{i,j}\omega_i\omega_j$ for each $i \in V_{K_1}^{-}$ and each $j \in V_{K_1}^{+}$. Provided that these values are known, we are able to provide conditions under which ω_i and ω_j exists and are less than one (Proposition 2 and Proposition 3), which allows to obtain the rates at which fair load balancing is achieved.

Let us consider the following algorithm to construct a feasible solution of the problem of determining the excess and deficit values of the nodes:

1. We create a heap with the values of $(\rho_{K_1} - \rho_i^{(I)})(\mu_i + \delta_i)$ for i in $V_{K_1}^{-}$.
2. We create another heap with the values $(-\rho_{K_1} + \rho_i^{(I)})(\mu_i + \delta_i)$ for j in $V_{K_1}^{+}$. [1]
3. We take the max of the first heap (say a) and the max of the second heap (say b). We remove both elements from their respective heaps. Let i be the index of a and j the index of b.
4. We consider $c = min(a, b)$. By construction $c > 0$. We affect c to the flow between i and j (i.e. $c = \rho_{K_1}\gamma_{i,j}\omega_i\omega_j$).
5. If $a - c > 0$ we add $a - c$ to the heap $V_{K_1}^{-}$ and we associate it with index i. Similarly if $b - c > 0$ we add $b - c$ to the heap $V_{K_1}^{+}$ and we associate it with index j.
6. If both heaps are non empty, we return to Step 3.

The main advantage of this simple method is its complexity. At each iteration we remove one or two nodes, using the ExtractMin operation to get the information from the heap, which requires $O(log(N))$ time (see Thm 10.1 in [18]). We find at most N values of $\rho_{K_1}\gamma_{i,j}\omega_i\omega_j$ for some i and j. At the end of this heuristic, we obtain the non zero values of $\rho_{K_1}\gamma_{i,j}\omega_i\omega_j$ for $i \in V_{K_1}^{-}$ and $j \in V_{K_1}^{+}$ with a worst complexity of $O(Nlog(N))$.

Remark 4. *Note that there might be simple scenarios where the polling rate to achieve load balancing can be obtained without computing the values of $\rho_{K_1}\gamma_{i,j}\omega_i\omega_j$. For instance, the model depicted in Figure 1(d), we only have two unbalanced nodes, in which case we have one node in excess and a deficit in the other node.*

Let us present an example to compute the deficit and excess values using the above heuristic:

Example 1. *Suppose that we have* 3 *cells in subset $V_{K_1}^{-}$ with deficit values* 1, 5, *and* 2 *(nodes number* 1, 2 *and* 3*) while we have* 4 *cells in $V_{K_1}^{+}$ (nodes number* 4, 5, 6 *and* 7*) with excess values (*2,1,1,4*). We will use* 4 *iterations:*

1. *In the first step, node 2 is the node with the largest deficit value of $V_{K_1}^{-}$ and its deficit value is 5, whereas node 7 is the one with the largest excess value of $V_{K_1}^{+}$ and its excess value is 4. Thus, we match these nodes for a value of* 4*. Node* 7 *is removed, the deficit value of node* 2 *becomes* 1*. Besides, we have that $\gamma_{2,7}\omega_2\omega_7 = 4$.*

[1]Note that we change the sign of the values to only deal with positive values.

2. *In this step, node 3 is the node with the largest deficit value of* $V^-_{K_1}$ *and the deficit value is 2, whereas the largest excess value of* $V^+_{K_1}$ *is in node* 4*, whose excess value is 2. Therefore, we match these nodes for a value of* 2 *and they are both removed. Besides, we have that* $\gamma_{3,4}\omega_3\omega_4 = 2$*.*
3. *In this step, there is a tie between nodes* 1 *and* 2*. In this case, we choose at random and, therefore, we match node* 1 *with node* 5 *for a value of* 1*. Both nodes are removed. Also, we have that* $\gamma_{1,5}\omega_1\omega_5 = 1$*.*
4. *Finally, we match node* 2 *with node* 6 *for a value of* 1*. Both nodes are removed. Also, we have that* $\gamma_{2,6}\omega_2\omega_6 = 1$*.*

We represent the obtained result of this example in Figure 5.

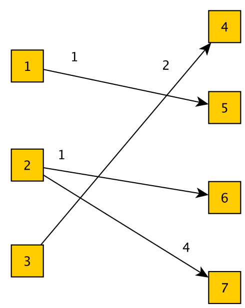


Figure 5: The solution leading to a fair load balancing using the considered method.

We remark that the values of $\rho_{K_1}\gamma_{i,j}\omega_i\omega_j$ for $i \in V^-_{K_1}$ and $j \in V^+_{K_1}$ are known. Therefore, for all $j \in V^+_{K_1}$, we denote by

$$d_j = \rho_{K_1} \sum_{i \in V^-_{K_1}} \gamma_{i,j}\omega_i\omega_j, \tag{6}$$

the values that we get at the end of the above heuristic. Note that d_j is already obtained while $\gamma_{i,j}$, ω_i, and ω_j are still unknown.

4.1.3. Numerical derivation of $\gamma_{i,j}$

We have obtained $\gamma_{i,j}\omega_i\omega_j$ in the previous section. In this section, we will analyze the values of ω_i and ω_j, whose computation allows us to obtain the rates $\gamma_{i,j}$. Taking into account that the cells of $V^+_{K_1}$ send DPs to the cells of $V^-_{K_1}$, for $i \in V^-_{K_1}$, we have:

$$\omega_i = \frac{\alpha_i}{\beta_i + \mu_i\rho_{K_1}\sum_{m=0}^{c_i-1}(\omega_i)^m + \sum_{j\in V^+_{K_1}}\gamma_{i,j}}, \tag{7}$$

while for $i \in V^+_{K_1}$, we have:

$$\omega_i = \frac{\alpha_i}{\beta_i + \mu_i\rho_{K_1}\sum_{m=0}^{c_i-1}(\omega_i)^m + \sum_{j\in V^-_{K_1}}\omega_j\gamma_{j,i}}. \tag{8}$$

We first focus on the solution of the former fixed point problem.

Proposition 2. *There exists a solution to* (8) *if and only if*

$$0 < \alpha_i - d_i < \beta_i + \mu_i \rho_{K_1} c_i.$$

Furthermore, if the solution exists, it is unique.

Proof. We get from (8) that

$$\omega_i \left(\beta_i + \mu_i \rho_{K_1} \sum_{m=0}^{c_i - 1} (\omega_i)^m + \sum_{j \in V_{K_1}^-} \omega_j \gamma_{j,i} \right) = \alpha_i \iff \alpha_i - d_i = \beta_i \omega_i + \mu_i \rho_{K_1} \sum_{m=1}^{c_i} (\omega_i)^m.$$

Let us introduce function $f_i(\omega_i) = -\alpha_i + d_i + \beta_i \omega_i + \mu_i \rho_{K_1} \sum_{m=1}^{c_i} (\omega_i)^m$, which is a polynomial function in ω_i such that $f'(\omega_i) > 0$. Therefore, function $f_i()$ is non decreasing in the interval $[0, 1]$ and $f_i(0) = d_i - \alpha_i$ and $f_i(1) = \beta_i + \mu_i \rho_{K_1} c_i + d_i - \alpha_i$. Therefore, the solution of (8) exists and is unique if and only if

$$0 < \alpha_i - d_i < \beta_i + \mu_i \rho_{K_1} c_i.$$

□

Assuming that the conditions of the above result hold for all the nodes in $V_{K_1}^+$, we can compute the value of ω_j for $j \in V_{K_1}^+$ with simple numerical methods, such as dichotomical search. Hence, we assume in the following that the value of ω_j for $j \in V_{K_1}^+$ is known. Therefore, we can obtain $\gamma_{i,j}\omega_i$ from $\gamma_{i,j}\omega_i\omega_j$. As a result, for all $i \in V_{K_1}^-$, we define $e_i = \sum_{j \in V_{K_1}^+} \gamma_{i,j}\omega_i$. Using the same arguments as in Proposition 2, we can provide necessary and sufficient conditions for the existence and uniqueness of a solution of the fixed point equation (7).

Proposition 3. *There exists a solution to* (7) *if and only if*

$$0 < \alpha_i - e_i < \beta_i + \mu_i \rho_{K_1} c_i.$$

Furthermore, if the solution exists, it is unique.

Finally, assuming that the conditions of the above result hold, we can obtain the value of ω_i and the value of ω_j by numerical methods and, as a result, we are able to compute the polling rates to achieved load balancing from $\gamma_{i,j}\omega_i\omega_j$.

4.1.4. Particular case: a network with a single block

In a network formed by a single block, we have that $K_1 = V$ and $\mu_i = 0$ for all i. Therefore, $\rho_{K_1} = \frac{\sum_{i=1}^N \lambda_i}{\sum_{i=1}^N \delta_i}$. Moreover, we can partition the set of nodes by V^+, V^- and $V^=$ and compute the excess and deficit values as previously. However, (7) and (8) for this case are given by

$$\omega_i = \frac{\alpha_i}{\beta_i + \sum_{j \in V^+} \gamma_{i,j}},$$

for $i \in V^-$, while for $i \in V^+$

$$\omega_i = \frac{\alpha_i}{\beta_i + \sum_{j\in V^-} \omega_j \gamma_{j,i}}.$$

From Proposition 2 and Proposition 3, the solution of the above expression exists if and only if $\alpha_i - e_i$ and $\alpha_i - d_i$ are positive and smaller than β_i. Moreover, when these conditions are satisfied, we have that $\omega_i = \frac{\alpha_i - e_i}{\beta}$ for all $i \in V^-$ and $\omega_i = \frac{\alpha_i - d_i}{\beta}$ for all $i \in V^+$.

4.2. Analyzing K_k by induction

We now assume that we have computed the polling rate to achieve load balancing for all layers $K_1, \ldots, K_{k-1}$. Note however that we do not have the same load ρ_i for DP queues which are not in the same layer. The fair load balancing is achieved within each layer with a possibly distinct load. By induction, the load ρ_i have been computed for all the DP queues in K_l for $l < k$. Therefore taking into account the assumptions about the topology, we have from (FLOW-DP) that:

$$(\mu_i + \delta_i)\rho_i = \lambda_i + \sum_{m=1}^{k-1} \sum_{j\in K_m} \mu_j \rho_j \omega_j^{c_j} P(j,i) + \sum_{j\in K_k} (\gamma_{i,j}\omega_i\omega_j\rho_i - \gamma_{j,i}\omega_j\omega_i\rho_j). \tag{9}$$

After summation on all the DP queues in K_k, the last term cancels as before, and we get:

$$\sum_{i\in K_k} (\mu_i + \delta_i)\rho_i = \sum_{i\in K_k} \lambda_i + \sum_{i\in K_k} \sum_{m=1}^{k-1} \sum_{j\in K_m} \mu_j \rho_j \omega_j^{c_j} P(j,i). \tag{10}$$

By induction, all the rates ρ_j and ω_j have already been computed as they are in a layer with a smaller index. Therefore the fair load balancing for the queues in K_k are defined by:

$$\rho_{K_k} = \frac{\sum_{i\in K_k} (\lambda_i + \sum_{m=1}^{k-1} \sum_{j\in K_m} \mu_j \rho_j \omega_j^{c_j} P(j,i))}{\sum_{i\in K_k} (\mu_i + \delta_i)} \tag{11}$$

Once this fair rate for DP queues has been computed, we process like for K_1 to obtain the energy rates for the EP queues in K_k and one can continue with the induction. Note that Remark 4 is still valid. For instance in Layer 2 of the graph depicted in Figure 4, we do not use the heuristics presented in the previous section to find $\gamma_{i,j}\omega_i\omega_j$ as we clearly have only one feasible solution for the load balancing.

5. Numerical Experiments

In this section, we present the numerical work we have carried out. We first explain how the load balancing protocol under consideration in this work is a good approach to improve the system performance. Then, we focus on the fair load balancing to find the polling rate required to achieve it. Finally, we study the influence of the parameters related to energy (leakage and arrival rates of EPs) on the fair load balancing polling rate. Throughout this section, we consider the network of Figure 2, which is formed by two layers and, in each layer, there are two cells. The solution of the fixed point equations we required to solve to determine the load of the EPs and DPs has been obtained by a simple fixed point heuristic, which converged in all the cases we have considered. Throughout this section, we consider that $A_{i,j}^{(1)} = 1$ when $\gamma_{i,j} > 0$, as in the previous section.

5.1. Improving Performance by Means of Load Balancing

We now focus on the performance improvement that can be achieved by using the load balancing technique we consider in this work. We will see that how to migrate jobs from cells which are close to saturation to cells with low load as well as the manner that this improves the performance of the system.

We consider the following values for the arrival rate of DPs to the cells: $\lambda_1 = 0.55$, $\lambda_2 = 0.99$, $\lambda_3 = 0.3$ and $\lambda_4 = 0.1$. Taking into account the topology of the network, we have that $\mu_3 = \mu_4 = 0$ as well as $\delta_1 = \delta_2 = 0$. Moreover, we consider the following values: $\mu_1 = \mu_2 = 1$ and $\delta_3 = \delta_4 = 1$. We also set $c_i = 1$ for all the cells $i = 1, 2, 3, 4$. For the arrival rate of EPs, we consider that $\alpha_i = 1.0$, $i = 1, 2, 3, 4$. We also consider that $\beta_i = 1.5$ for $i = 1, 2, 3, 4$.

Regarding the movement of jobs due to load balancing, we have that DP queue 1 (resp. DP queue 4) receives DP packets from DP queue 2 (resp. DP queue 3). Finally, we have that $\gamma_{1,2} = \gamma_{4,3} = \gamma$. Our goal is to study the influence of γ on the system performance.

We first note that, when $\gamma = 0$, the load of DP queue 2 and of DP queue 3 are very close to one. As a consequence, we have that the mean number of DPs (and, from Little's Law, the mean response time as well) of these queues is very large. In Figure 6 we plot the evolution of the loads of the DPs of all the cells when γ varies from 0 to 100. We observe that with small values of γ the load of DP queue 2 and of DP queue 3 decrease substantially. In fact, when $\gamma \geq 11$, the load of all the DPs is smaller than 0.8, which implies that none of the DP queues are in saturation with small values of γ.

We have measured the improvement of the performance as a result of the load balancing technique by the mean number of customers when $\gamma = 0$ and $\gamma = 50$. When $\gamma = 0$, we have that the mean number of customers of DP queue 2 is equal to $0.99//(1 - 0.99) = 99$. However, when $\gamma = 50$, we have that the load of DP queue 2 is equal to 0.77 and, as a consequence, the mean number of customers of this queue is $0.77/(1 - 0.77) = 2.33$, which is very small comparing with 99. A similar conclusion can be derived when we consider the load of the DP queue 3. Hence, from this figure, we conclude that the mean number of customers can be decreased a lot with relatively small values of the polling rate of the load balancing.

An interesting property of Figure 6 is that, when γ is large, the influence of the load balancing is very small. Indeed, as it can be seen in Figure 7, ω_1 and γ_3, i.e., the load of EPs of cell 1 and of cell 3, are smaller than 0.02 when γ is larger than 50. This means that when γ is large, there is no energy to perform load balancing (i.e., (COND1) or (COND2) are not satisfied) and, therefore, we cannot migrate jobs from some cells to others even though we increase the polling rate. We also conclude from the illustration of Figure 7 that for the values of γ we have considered, all the EP queues are stable, i.e. the load of EPs is strictly smaller than one in all the cells.

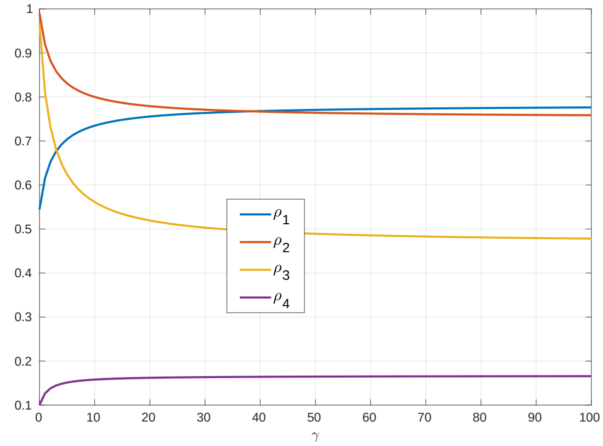


Figure 6: The evolution of ρ_1, ρ_2, ρ_3 and ρ_4 with respect to γ.

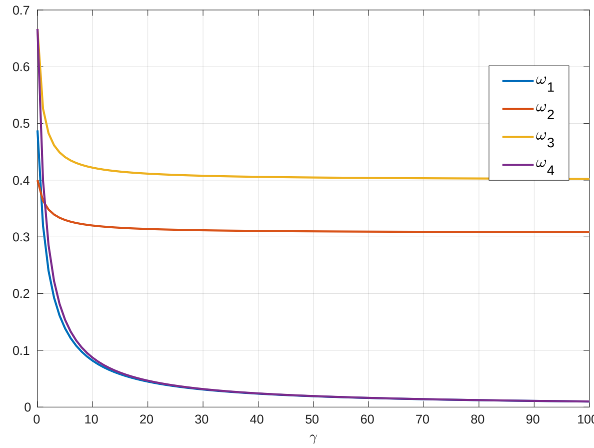


Figure 7: The evolution of ω_1, ω_2, ω_3 and ω_4 with respect to γ.

5.2. Analysis of Load Balancing Rates

In Section 4, we have obtained a method to compute the polling rate to achieve fair load balancing in each of the layers of a network. In this section, we provide an alternative numerical method to compute this polling rate.

We first focus on the first layer of the network and we aim to determine how DPs must be moved so that it is ensured that the load of DPs of cell 1 and of cell 2 are the same. We consider the following parameters: $\lambda_1 = 0.2$, $\lambda_2 = 0.3$, $\mu_1 = \mu_2 = 1$, $\delta_1 = \delta_2 = 0$, $\alpha_1 = 0.5$, $\alpha_2 = 2$ and $\beta_1 = \beta_2 = 2$.

We first note that, without load balancing, the load of DPs of cell 1 is 0.3, whereas the load of DPs in cell 2 is 0.2. This means that the migration of DPs in this case must be from cell 1 to cell 2 or, in other words, we have that $\gamma_{1,2} > 0$ and $\gamma_{2,1} = 0$. Furthermore, we know that, when the fair load balancing is achieved, the load of DPs of cell 1 and of cell 2 is equal to 0.25.

In Figure 8 we present the evolution of ρ_1 and ρ_2 when $\gamma_{1,2}$ varies from 0 to 7.5. We see that the fair load balancing in the first layer is achieved for $\gamma_{12} \approx 6.75$. We also study the evolution of ω_1 and ω_2 for these values of $\gamma_{1,2}$ to see whether these queues are stable for these instances and, according to the plot of Figure 9, we conclude that the load of the EPs in cell 1 and cell 2 is smaller than one for all the considered values of γ_{12}.

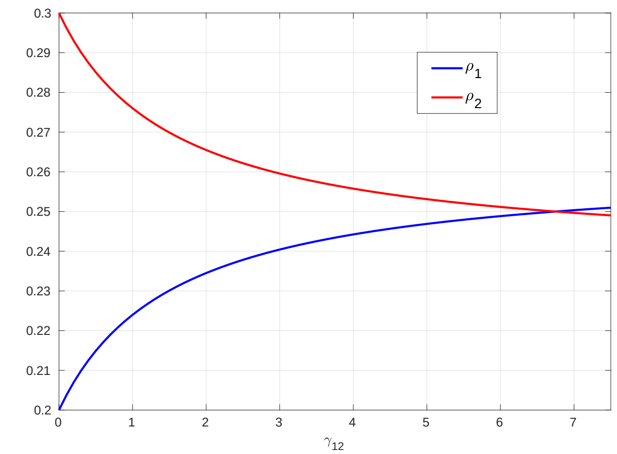


Figure 8: Evolution of ρ_1 and ρ_2 when γ_{12} varies from 0 to 7.5.

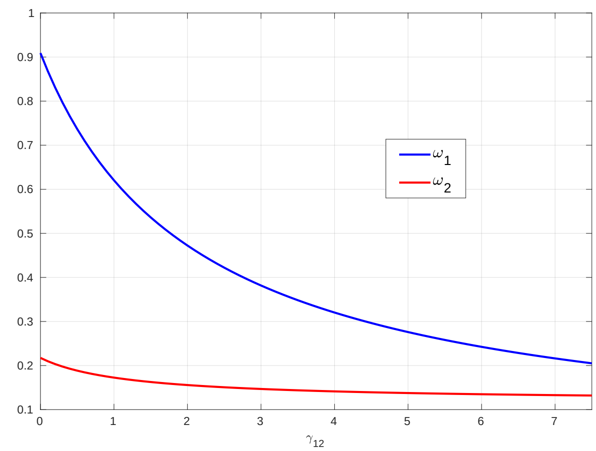


Figure 9: Evolution of ω_1 and ω_2 when γ_{12} varies from 0 to 7.5.

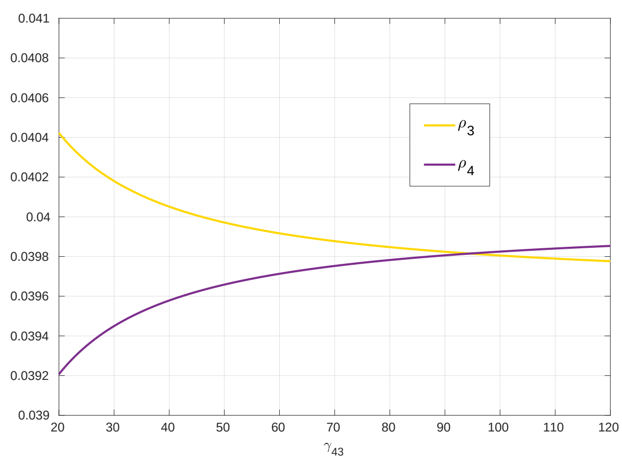


Figure 10: Evolution of ρ_3 and ρ_4 when γ_{43} varies from 20 to 120.

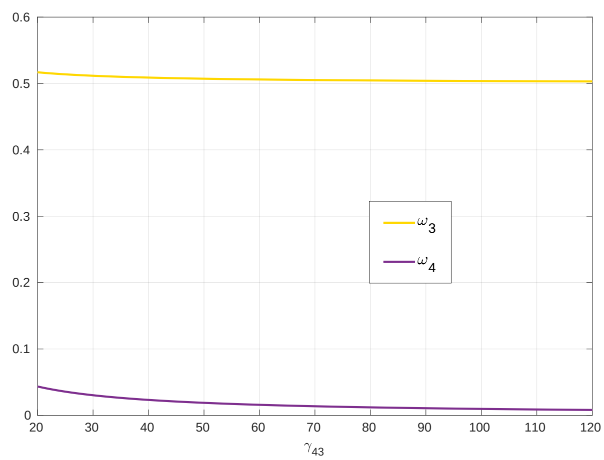


Figure 11: Evolution of ω_1 and ω_2 when γ_{43} varies from 20 to 120.

We now study the polling rate at which fair load balancing is achieved in the second layer of the network, that is, when the load of DPs in cell 3 and cell 4 are the same. According the our previous analysis, we consider that $\gamma_{1,2} = 6.75$. We also consider the following parameters: $\lambda_3 = 0.05$, $\lambda_4 = 0.1$, $\mu_3 = \mu_4 = 0$, $\delta_3 = \delta_4 = 3$, $\alpha_3 = 2$, $\alpha_4 = 1$ and $\beta_3 = \beta_4 = 3$. We also notice from the network topology that $P(1,3) = 1$ and $P(3,2) = 1$.

We have first computed the load of DPs in cell 3 and cell 4 when the polling rate is zero and we obtained that $\rho_3 = 0.046$ and $\rho_4 = 0.033$. This means that $\gamma_{4,3} > 0$ and $\gamma_{3,4} = 0$, i.e., in the load balancing protocol we are transferring DPs from cell 3 to cell 4.

We study in Figure 10, the evolution of ρ_3 and ρ_4 when $\gamma_{4,3}$ varies from 20 to 120. We observe that the value of the polling rate at which fair load balancing is achieved in this instance is $\gamma_{43} \approx 94.5$. We also show in Figure 11 the evolution of ω_3 and ω_4 for different values of $\gamma_{4,3}$ and we observe that for all the considered values of the polling rate the EP queues are stable, i.e., $\omega_3 < 1$ and $\omega_4 < 1$.

5.3. Influence of Energy Parameters for Fair Load Balancing

In this section, we analyze the effect of the leakage rate and the arrival rate of EPs, which are the parameters that influence more the load of the EPs, in the polling rate required to achieve load balancing. In the experiments that we describe in this section we consider the first layer of the network presented in Figure 2 and the same parameters as in the previous section. We will vary one of the parameters in each experiment to see its impact in the polling rate to achieve load balancing.

We now focus on the leakage rate of the EPs of cells 1 and 2. We first analyze the case where the leakage rate of cell 1 is large. To this end, we consider $\beta_1 = 20$ and the rest of the parameters as in the previous section. We illustrate in Figure 12 the evolution of the loads of the DPs of cell 1 and cell 2 with respect to $\gamma_{1,2}$. We consider that $\gamma_{1,2}$ varies from 0 to 80. We observe from this plot that the polling rate to achieve load balancing is equal to 60.25, which is much larger than the polling rate when $\beta_1 = 2$ (which is 6.75, according to Figure 8). From this figure, we conclude that increasing the leakage rate of the receiver cell, i.e., of cell 1, requires a larger polling rate to achieve fair load balancing. We observe a similar behavior when we increase the leakage rate of cell 2 in Figure 13. Indeed, in this case, we consider that $\beta_2 = 2.6$ and the rest of the parameters are fixed to those of the experiments of the previous section. In fat, in Figure 13, we present the evolution of the loads of DPs of cell 1 and cell 2 when $\gamma_{1,2}$ varies from 0 to 50 and we observe that the polling rate to achieve load balancing in this case is approximately equal to 42.75, which is also much higher than 6.75 but not as high as 60.25. Therefore, we conclude that the polling rate to achieve load balancing increases more when we vary the leakage rate of the first cell.

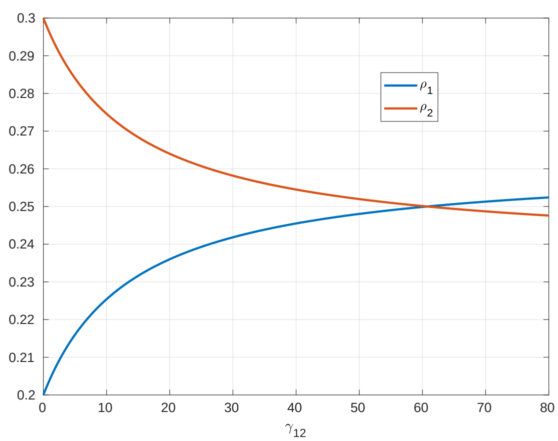


Figure 12: Evolution of ρ_1 and ρ_2 when γ_{12} varies from 0 to 80 and $\beta_1 = 20$

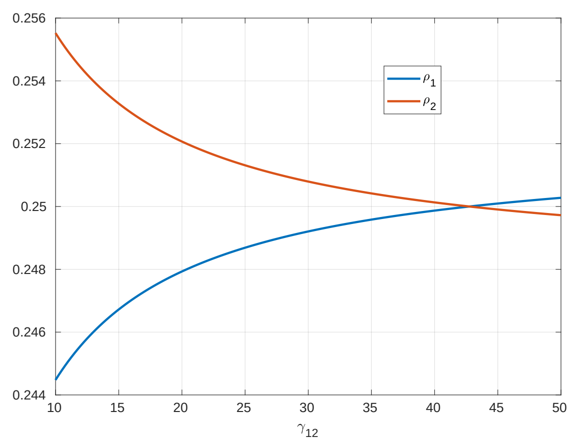


Figure 13: Evolution of ρ_1 and ρ_2 when γ_{12} varies from 10 to 50 and $\beta_2 = 2.6$.

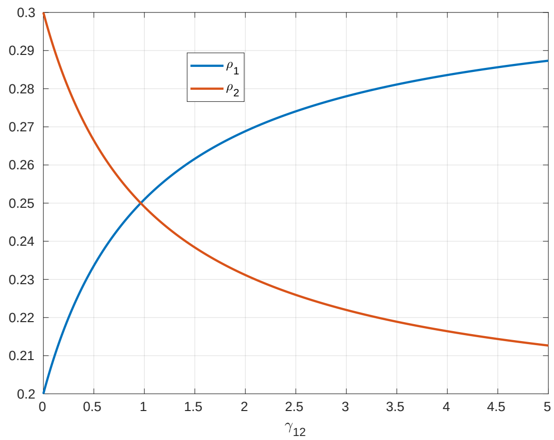


Figure 14: Evolution of ρ_1 and ρ_2 when γ_{12} varies from 0 to 5 and $\alpha_1 = 5$

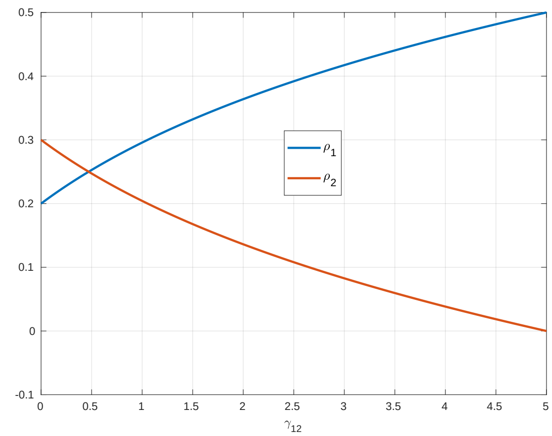


Figure 15: Evolution of ρ_1 and ρ_2 when γ_{12} varies from 0 to 5 and $\alpha_2 = 1.5$.

We also explore the influence on the polling rate for fair load balancing when we increase the arrival rate of EPs to these cells. First, we consider in Figure 14 that α_1 is equal to 5 and the rest of the parameters are as in the previous section. We plot the evolution of the load of DPs in cell 1 and cell 2 with respect to $\gamma_{1,2}$. We consider that $\gamma_{1,2}$ varies from 0 to 5. We observe that the polling rate to achieve fair load balancing in this case is approximately equal to 0.96, which is clearly smaller than 6.75. Therefore, we conclude from this experiment that the polling rate required to achieve load balancing decreases with α_1, i.e., with the arrival rate of EPs in the receiver cell. We also analyze the influence of the arrival rate of EPs of the sender cell. We consider that $\alpha_2 = 1.5$ in this case and in Figure 15 we plot the evolution of ρ_1 and ρ_2 with respect to $\gamma_{1,2}$. We consider that $\gamma_{1,2}$ varies from 0 to 5. We observe that the polling rate to achieve load balancing in this case is approximately equal to 0.46, which is smaller than 6.75 and also smaller than 0.96. From this experiment we conclude that the polling rate required to achieve load balancing also decreases with α_2 and also that the arrival rate of the sender cell, i.e. α_2, influences more this polling rate.

5.4. A single block with 4 cells

We consider a network formed by a single block with 4 cells. For this case, we use the methodology presented in Section 4 to compute the polling rates required to achieve fair load balancing. Given that there is a single block, we have that $\mu_i = 0$ for all i. We consider $\delta_i = 1$ for all i and, regarding the arrival rate of DPs, $\lambda_1 = 0.65$, $\lambda_2 = 0.5$, $\lambda_3 = 0.35$, and $\lambda_4 = 0.3$. On the other hand, for EPs, we consider that $\beta_i = 1$ for all i and $\alpha_1 = 0.3$, $\alpha_2 = 0.1$, $\alpha_3 = 4$ and $\alpha_4 = 4$.

We first compute ρ_{K_1} for this network as follows

$$\rho_{K_1} = \frac{0.65 + 0.5 + 0.35 + 0.3}{1 + 1 + 1 + 1} = 0.45.$$

Therefore, since $\rho_1^{(I)} = 0.65$, $\rho_2^{(I)} = 0.5$, $\rho_1^{(I)} = 0.35$ and $\rho_1^{(I)} = 0.3$, we have that $V_{K_1}^+ = \{1, 2\}$ and $V_{K_1}^- = \{3, 4\}$, which means that cells 1 and 2 send DPs to cells 3 and 4. Hence, all the polling rates that are different from $\gamma_{3,1}, \gamma_{4,1}, \gamma_{3,2}$ and $\gamma_{4,2}$ are equal to zero. We now aim to compute the exact value of these polling rates.

The next step consists of finding the excess and deficit values of the cells. For cell 1, we obtain that $(\rho_{K_1} - \rho_1^{(I)})(\mu_1 + \delta_1) = 0.2$. In an analogous manner, we get that $(\rho_{K_1} - \rho_2^{(I)})(\mu_2 + \delta_2) = 0.05$, $(-\rho_{K_1} + \rho_3^{(I)})(\mu_3 + \delta_3) = 0.1$ and $(-\rho_{K_1} + \rho_4^{(I)})(\mu_4 + \delta_4) = 0.15$.

Using the heuristic presented in Section 4.1.2, we obtain that cell 1 sends DPs to cell 3 and 4, whereas cell 2 sends DPs only to cell 3 (which means that, in our solution, $\gamma_{4,2} = 0$ as well as $\gamma_{3,1} > 0, \gamma_{3,2} > 0$ and $\gamma_{4,1} > 0$). Besides, this algorithm provides us the following excess and deficit values:

$$\rho_{K_1}\gamma_{4,1}\omega_4\omega_1 = 0.15, \rho_{K_1}\gamma_{3,1}\omega_3\omega_1 = 0.05, \rho_{K_1}\gamma_{3,2}\omega_2\omega_3 = 0.05.$$

We now compute d_1 and d_2 in the following way:

$$d_1 = \rho_{K_1}\gamma_{4,1}\omega_4\omega_1 + \rho_{K_1}\gamma_{3,1}\omega_3\omega_1 = 0.15 + 0.05 = 0.2,$$

$$d_2 = \rho_{K_1}\gamma_{3,2}\omega_3\omega_2 = 0.05.$$

We now compute ω_1 and we observe that $\alpha_1 - d_1 = 0.1$, which is positive and smaller that β_1 (which is equal to one). Therefore, according to Proposition 2, there is a single solution, which is given by $\omega_1 = \frac{\alpha_1 - d_1}{\beta_1} = 0.1$ because there is a single block (i.e., $\mu_1 = 0$). In an analogous manner, we obtain that $\omega_2 = \frac{\alpha_2 - d_2}{\beta_2} = 0.05$. From these results, we obtain that

$$\gamma_{4,1}\omega_4 = \frac{0.15}{\rho_{K_1}\omega_1} = \frac{0.15}{0.45 \cdot 0.1} = \frac{10}{3},$$

$$\gamma_{3,1}\omega_3 = \frac{0.05}{\rho_{K_1}\omega_1} = \frac{0.05}{0.45 \cdot 0.1} = \frac{10}{9},$$

and

$$\gamma_{3,2}\omega_3 = \frac{0.05}{\rho_{K_1}\omega_2} = \frac{0.05}{0.45 \cdot 0.05} = \frac{20}{9}.$$

We compute the values of e_3 and e_4 as follows:

$$e_3 = \gamma_{3,1}\omega_3 + \gamma_{3,2}\omega_3 = \frac{10}{9} + \frac{20}{9} = \frac{10}{3},$$

and

$$e_4 = \gamma_{4,1}\omega_4 = \frac{10}{3}.$$

The above values are used to compute ω_3 and ω_4. We first focus on ω_3, which, taking into account that $\alpha_3 - e_3 = \frac{2}{3}$ is positive and smaller than β_3, we know that it has a single solution from Proposition 3, which is given by $\omega_3 = \frac{\alpha_3 - e_3}{\beta_3} = \frac{2}{3}$. Likewise, we have that $\alpha_4 - e_4 = \frac{2}{3}$, which applying again Proposition 3 we get that $\omega_4 = \frac{\alpha_4 - e_4}{\beta_4} = \frac{2}{3}$.

Finally, we proceed as follows to obtain the polling rates to achieve fair load balancing:

$$\rho_{K_1}\gamma_{4,1}\omega_4\omega_1 = 0.15 \iff \gamma_{4,1} = \frac{0.15}{\rho_{K_1}\omega_4\omega_1} = \frac{0.15}{0.45 \cdot \frac{2}{3} \cdot 0.1} = 5,$$

$$\rho_{K_1}\gamma_{3,1}\omega_3\omega_1 = 0.05 \iff \gamma_{3,1} = \frac{0.05}{\rho_{K_1}\omega_3\omega_1} = \frac{0.05}{0.45\cdot\frac{2}{3}\cdot 0.1} = \frac{5}{3},$$

and

$$\rho_{K_1}\gamma_{3,2}\omega_3\omega_2 = 0.05 \iff \gamma_{3,2} = \frac{0.05}{\rho_{K_1}\omega_3\omega_2} = \frac{0.05}{0.45\cdot\frac{2}{3}\cdot 0.05} = \frac{10}{3}.$$

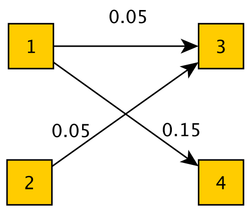


Figure 16: The load balancing graph when $\lambda_1 = 0.65$, $\lambda_2 = 0.5$, $\lambda_3 = 0.35$, and $\lambda_4 = 0.3$.

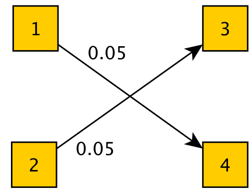


Figure 17: The load balancing graph when $\lambda_1 = 0.65$, $\lambda_2 = 0.5$, $\lambda_3 = 0.4$, and $\lambda_4 = 0.25$.

In this example, we have obtained that cell 1 sends DPs to cell 3 and 4, whereas cell 2 sends DPs only to cell 3, that is, the load balancing graph is given in Figure 16. We now consider that $\lambda_3 = 0.4$ and $\lambda_4 = 0.25$ and the rest of the parameters as in the previous case. We observe that $\rho_{K_1} = 0.45$ and, therefore, the sets $V_{K_1}^+$ and $V_{K_1}^-$ are the same as in the previous example. However, the load balancing graph is different since cell 1 sends DPs only to cell 4 and cell 2 sends DPs only to cell 3, see Figure 17. Proceeding in the same way as in the previous example, we obtain that $\gamma_{4,1} = 5$ and $\gamma_{3,2} = \frac{5}{4}$.

6. Conclusions

We analyzed an extension of the EPN model that allows migration of jobs from DP queues of different cells. That is, a batch of jobs are transferred from one queue to an idle queue when there is enough energy. We showed that the steady-state distribution of jobs in the queues has a product form expression when a solution of a fixed point equation exists. We provide sufficient conditions for the existence of a solution of the fixed point equation. In a layered network, we study how the polling rates (i.e. the rate at which a queue request to transfer DPs to another queue) must be set so as to ensure a fair load balancing, i.e., to ensure that the load of the DPs of the cells that are in the same layer is equal. Using numerical experiments, we show that dynamic load balancing can be very helpful to improve performance when there are DP queues that are in heavy-traffic. We also analyze numerically the polling rates to achieve a fair load balancing.

For future work, we are interested in studying variants of this model which might more interesting from a practical point of view; for instance, we will explore more complex conditions on the load balancing than those considered in this article and we will relax the assumption that the energy consumed for load balancing is independent of the number of migrated jobs. We would also like to analyze the polling rates of layered networks by considering other objective functions, such as minimizing energy consumption. We are also planning to study load balancing protocols in other EPN networks such as considering generally distributed service requirements [19] or finite buffer in the batteries [20]. We think that an interesting researchline is to consider the extension of the quasireversibility theory to accommodate signals [21] to analyze the existence of the product-form solution of this and other EPN models.

References

[1] D. L. Eager, E. D. Lazowska, J. Zahorjan, A comparison of receiver-initiated and sender-initiated adaptive load sharing, Performance evaluation 6 (1) (1986) 53–68.

[2] W. Minnebo, B. Van Houdt, A fair comparison of pull and push strategies in large distributed networks, IEEE/ACM Transactions on Networking 22 (3) (2013) 996–1006.

[3] N. Gast, G. Bruno, A mean field model of work stealing in large-scale systems, ACM SIGMETRICS Performance Evaluation Review 38 (1) (2010) 13–24.

[4] N. Sonenberg, G. Kielanski, B. Van Houdt, Performance analysis of work stealing in large-scale multithreaded computing, ACM Transactions on Modeling and Performance Evaluation of Computing Systems 6 (2) (2021) 1–28.

[5] T. He, S. Chen, H. Kim, L. Tong, K.-W. Lee, Scheduling parallel tasks onto opportunistically available cloud resources, in: 2012 IEEE Fifth International Conference on Cloud Computing, 2012, pp. 180–187. doi:10.1109/CLOUD.2012.15.

[6] M. S. Squillante, R. D. Nelson, Analysis of task migration in shared-memory multiprocessor scheduling, ACM SIGMETRICS Performance Evaluation Review 19 (1) (1991) 143–155.

[7] A. Marin, S. Balsamo, J. Fourneau, LB-networks: A model for dynamic load balancing in queueing networks, Perform. Eval. 115 (2017) 38–53.

[8] E. Gelenbe, Energy packet networks: Ict based energy allocation and storage, GreeNets 2011 (2011).

[9] E. Gelenbe, Energy packet networks: smart electricity storage to meet surges in demand, in: Proceedings of the 5th International ICST Conference on Simulation Tools and Techniques, 2012, pp. 1–7.

[10] E. Gelenbe, A sensor node with energy harvesting., SIGMETRICS Performance Evaluation Review 42 (2) (2014) 37–39.

[11] E. Gelenbe, Product-form queueing networks with negative and positive customers, Journal of applied probability 28 (3) (1991) 656–663.

[12] E. Gelenbe, G-networks with instantaneous customer movement, Journal of Applied Probability 30 (3) (1993) 742–748.

[13] E. Gelenbe, G-networks with signals and batch removal, Probability in the Engineering and Informational Sciences 7 (3) (1993) 335–342.

[14] O. H. Abdelrahman, E. Gelenbe, A diffusion model for energy harvesting sensor nodes, in: 24th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, MASCOTS 2016, London, United Kingdom, IEEE Computer Society, 2016, pp. 154–158.

[15] P. P. Ray, Energy packet networks: an annotated bibliography, SN Computer Science 1 (1) (2020) 6.

[16] T.-H. Dao Thi, J.-M. Fourneau, Markovian queueing network with complex synchronizations: Product form and tensor, Performance Evaluation 69 (3) (2012) 164–186.

[17] P. Brémaud, Markov Chains: Gibbs fields, Monte Carlo Simulation and Queues, Springer-Verlag, 1999.

[18] T. Roughgarden, Algorithms Illuminated: part 2, Graph Algorithms and Data Structures, Cambridge University Press, 2022.

[19] Y. A. El Mahjoub, J.-M. Fourneau, H. Castel-Taleb, Energy packet networks with general service time distribution, in: 2020 28th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), IEEE, 2020, pp. 1–8.

[20] S. Samain, J. Doncel, A. Busic, J.-M. Fourneau, Multiclass energy packet networks with finite capacity energy queues, Performance Evaluation 152 (2021) 102228.

[21] X. Chao, M. Miyazawa, M. Pinedo, Queueing Networks Customers, Signals and Product Form solutions, John Wiley & Sons, 1999.

[22] C. D. Garcia, W. I. Zangwill., Pathways to solutions, fixed points, and equilibria, Prentice-Hall, Englewood Cliffs, N.J., 1981.

Appendix A. Some properties of the distribution of Assumption 1

We now present some properties of $B_{i,j}(X_j)$, the number of customers that move from cell j to cell i when there are X_j DPs at cell j. We first show that it is, indeed, a distribution for all X_0. To prove this, we proceed as follows:

$$\sum_{k=0}^{X_j-1}(1-b_{i,j})b_{i,j}^k + b_{i,j}^{X_j} = (1-b_{i,j})\sum_{k=0}^{X_j-1} b_{i,j}^k + b_{i,j}^{X_j} = (1-b_{i,j})\frac{1-b_{i,j}^{X_j}}{(1-b_{i,j})} + b_{i,j}^{X_j} = 1 - b_{i,j}^{X_j} + b_{i,j}^{X_j} = 1.$$

We now provide the expression of the expected value of $B_{i,j}(X_j)$:

$$\mathbb{E}[B_{i,j}(X_j)] = X_j b_{i,j}^{X_j} + \sum_{k=1}^{X_j-1} k(1-b_{i,j})b_{i,j}^k = X_j b_{i,j}^{X_j} + \sum_{k=1}^{X_j-1} kb_{i,j}^k - \sum_{k=1}^{X_j-1} kb_{i,j}^{k+1} = \sum_{k=1}^{X_j} kb_{i,j}^k - \sum_{k=1}^{X_j}(k-1)b_{i,j}^k = \sum_{k=1}^{X_j} b_{i,j}^k$$

From this expression, we conclude that $\mathbb{E}[B_{i,j}(X_j)]$ is increasing with X_j. This means that the more customers present in the queue, the more customers (in expectation) are sent during the load balancing. From this monotonicity, we can provide the following upper-bound:

$$\mathbb{E}[B_{i,j}(X_j)] < \mathbb{E}[B_{i,j}(\infty)] = \frac{b_{i,j}}{1-b_{i,j}}.$$

This means that the expected value of the number of customers that move from cell j to cell i during the load balancing is, at most, $\frac{b_{i,j}}{1-b_{i,j}}$.

Appendix B. Proof of Theorem 1

Recall that $B_{i,j}^{(m)}(X_j)$ is the probability of moving m customers due to the load balancing from DP queue j to DP queue i when there are X_j DPs at cell j. We have clearly that, for all X_j, $\sum_{m=0}^{X_j} B_{i,j}^{m}(X_j) = 1$. We now write the global balance equations:

$$\begin{aligned}
\pi(X,Y)\Big(\textstyle\sum_i \lambda_i \quad & + \textstyle\sum_i \alpha_i + \sum_i \delta_i 1_{X_i>0} + \sum_i \mu_i 1_{X_i>0} + \sum_i \beta_i 1_{Y_i>0} + \sum_i \sum_j \gamma_{i,j} 1_{Y_i>0}\Big) = \\
& \textstyle\sum_i \pi(X-e_i,Y)\lambda_i 1_{X_i>0} \\
+ \quad & \textstyle\sum_i \pi(X,Y-e_i)\alpha_i 1_{Y_i>0} \\
+ \quad & \textstyle\sum_i \pi(X,Y+e_i)\beta_i \\
+ \quad & \textstyle\sum_i \pi(X+e_i,Y)\delta_i \\
+ \quad & \textstyle\sum_i \sum_{m=0}^{c_i-1} \pi(X+e_i,Y+me_i)\mu_i 1_{Y_i=0} \\
+ \quad & \textstyle\sum_i \sum_j \pi(X+e_i-e_j,Y+c_ie_i)\mu_i P(i,j) 1_{X_j>0} \\
+ \quad & \textstyle\sum_i \sum_j \sum_{k\geq 1} \sum_{l=0}^{k-1} A_{i,j}^{(k)} \pi(X,Y+e_i+le_j)\gamma_{i,j} 1_{Y_j=0} \\
+ \quad & \textstyle\sum_i \sum_j \sum_{k\geq 0} A_{i,j}^{(k)} \pi(X,Y+e_i+ke_j)\gamma_{i,j} 1_{X_i>0} \\
+ \quad & \textstyle\sum_i \sum_j \sum_{m\geq 0} \sum_{k\geq 0} A_{i,j}^{(k)} \gamma_{i,j} B_{i,j}^{(m)}(X_j+m)\pi(X-e_im+e_jm,Y+e_i+ke_j) 1_{X_i=m}
\end{aligned}$$

The lhs of the above expression represents the total flow out from state (X,Y) and the rhs the total flow into (X,Y). The rhs of the above expression is formed by nine terms (each of them is represented in a different line). The first sum represents the arrival of a DP and the second one of the EP. The first sum represents the leakage of an EP and the fourth one the departure of a DP. The fifth sum represents that the lost of a DP after being served because the required energy is not available. The sixth sum represents the successful movement of a DP after getting service to the next cell. The seventh sum represents that (COND2) is not satisfied, whereas the eighth sum that (COND3) is not satisfied. Finally, the last sum represent a successful load balancing.

We divide by $\pi(X,Y)$ the above expression and, assuming that the expression in (1) is true, we get that:

$$
\begin{aligned}
\sum_i \lambda_i + \sum_i \alpha_i + \quad & \sum_i \delta_i 1_{X_i>0} + \sum_i \mu_i 1_{X_i>0} + \sum_i \beta_i 1_{Y_i>0} + \sum_i \sum_j \gamma_{i,j} 1_{Y_i>0} = \\
& \sum_i \frac{\lambda_i}{\rho_i} 1_{X_i>0} \\
+ \quad & \sum_i \frac{\alpha_i}{\omega_i} 1_{Y_i>0} \\
+ \quad & \sum_i \beta_i \omega_i \\
+ \quad & \sum_i \rho_i \delta_i \\
+ \quad & \sum_i \sum_{m=0}^{c_i-1} \rho_i \omega_i^m \mu_i 1_{Y_i=0} \\
+ \quad & \sum_i \sum_j \frac{\rho_i}{\rho_j} \omega_i^{c_i} \mu_i P(i,j) 1_{X_j>0} \\
+ \quad & \sum_i \sum_j \sum_{k\geq 1} \sum_{l=0}^{k-1} A_{i,j}^{(k)} \omega_i \omega_j^l \gamma_{i,j} 1_{Y_j=0} \\
+ \quad & \sum_i \sum_j \sum_{k\geq 0} A_{i,j}^{(k)} \omega_i \omega_j^k \gamma_{i,j} 1_{X_i>0} \\
+ \quad & \sum_i \sum_j \sum_{m\geq 0} \sum_{k\geq 0} A_{i,j}^{(k)} \gamma_{i,j} B_{i,j}^{(m)} (X_j + m) \left(\frac{\rho_j}{\rho_i}\right)^m \omega_i \omega_j^k 1_{X_i=m}
\end{aligned}
$$

We use that $1_{Y_i=0} = 1 - 1_{Y_i>0}$ and $1_{Y_j=0} = 1 - 1_{Y_j>0}$ and put the negative terms to the other side:

$$
\begin{aligned}
\sum_i \lambda_i + \sum_i \alpha_i + \quad & \sum_i \delta_i 1_{X_i>0} + \sum_i \mu_i 1_{X_i>0} + \sum_i \beta_i 1_{Y_i>0} + \sum_i \sum_j \gamma_{i,j} 1_{Y_i>0} + \sum_i \sum_{m=0}^{c_i-1} \rho_i \omega_i^m \mu_i 1_{Y_i>0} \\
+ \quad & \sum_i \sum_j \sum_{k\geq 1} \sum_{l=0}^{k-1} A_{i,j}^{(k)} \omega_i \omega_j^l \gamma_{i,j} 1_{Y_j>0} \\
= \quad & \sum_i \frac{\lambda_i}{\rho_i} 1_{X_i>0} \\
+ \quad & \sum_i \frac{\alpha_i}{\omega_i} 1_{Y_i>0} \\
+ \quad & \sum_i \beta_i \omega_i \\
+ \quad & \sum_i \rho_i \delta_i \\
+ \quad & \sum_i \sum_{m=0}^{c_i-1} \rho_i \omega_i^m \mu_i \\
+ \quad & \sum_i \sum_j \frac{\rho_i}{\rho_j} \omega_i^{c_i} \mu_i P(i,j) 1_{X_j>0} \\
+ \quad & \sum_i \sum_j \sum_{k\geq 1} \sum_{l=0}^{k-1} A_{i,j}^{(k)} \omega_i \omega_j^l \gamma_{i,j} \\
+ \quad & \sum_i \sum_j \sum_{k\geq 0} A_{i,j}^{(k)} \omega_i \omega_j^k \gamma_{i,j} 1_{X_i>0} \\
+ \quad & \sum_i \sum_j \sum_{m\geq 0} \sum_{k\geq 0} A_{i,j}^{(k)} \gamma_{i,j} B_{i,j}^{(m)} (X_j + m) \left(\frac{\rho_j}{\rho_i}\right)^m \omega_i \omega_j^k 1_{X_i=m}
\end{aligned}
$$

We now rewrite the last term of the last equation in the following equivalent form (in which we differentiate the case where the DP queue of the sender gets empty after the load balancing and not):

$$
\begin{aligned}
\sum_i \lambda_i + \sum_i \alpha_i + \quad & \sum_i \delta_i 1_{X_i>0} + \sum_i \mu_i 1_{X_i>0} + \sum_i \beta_i 1_{Y_i>0} + \sum_i \sum_j \gamma_{i,j} 1_{Y_i>0} + \sum_i \sum_{m=1}^{c_i-1} \rho_i \omega_i^m \mu_i 1_{Y_i>0} \\
+ \quad & \sum_i \sum_j \sum_{k\geq 1} \sum_{l=0}^{k-1} A_{i,j}^{(k)} \omega_i \omega_j^l \gamma_{i,j} 1_{Y_j>0} \\
= \quad & \sum_i \frac{\lambda_i}{\rho_i} 1_{X_i>0} \\
+ \quad & \sum_i \frac{\alpha_i}{\omega_i} 1_{Y_i>0} \\
+ \quad & \sum_i \beta_i \omega_i \\
+ \quad & \sum_i \rho_i \delta_i \\
+ \quad & \sum_i \sum_{m=0}^{c_i-1} \rho_i \omega_i^m \mu_i \\
+ \quad & \sum_i \sum_j \frac{\rho_i}{\rho_j} \omega_i^{c_i} \mu_i P(i,j) 1_{X_j>0} \\
+ \quad & \sum_i \sum_j \sum_{k\geq 1} \sum_{l=0}^{k-1} A_{i,j}^{(k)} \omega_i \omega_j^l \gamma_{i,j} \\
+ \quad & \sum_i \sum_j \sum_{k\geq 0} A_{i,j}^{(k)} \omega_i \omega_j^k \gamma_{i,j} 1_{X_i>0} \\
+ \quad & \sum_i \sum_j \sum_{m\geq 0} \sum_{k\geq 0} A_{i,j}^{(k)} \gamma_{i,j} B_{i,j}^{(m)}(X_j+m) \left(\frac{\rho_j}{\rho_i}\right)^m \omega_i \omega_j^k 1_{X_i=m} 1_{X_j>0} \\
+ \quad & \sum_i \sum_j \sum_{m\geq 0} \sum_{k\geq 0} A_{i,j}^{(k)} \gamma_{i,j} B_{i,j}^{(m)}(m) \left(\frac{\rho_j}{\rho_i}\right)^m \omega_i \omega_j^k 1_{X_i=m} 1_{X_j=0}
\end{aligned}
$$

From Assumption 1, we can write $B_{i,j}^{(m)}(m)\left(\frac{\rho_j}{\rho_i}\right)^m = 1$ and $B_{i,j}^{(m)}(X_j+m)\left(\frac{\rho_j}{\rho_i}\right)^m = 1 - \frac{\rho_i}{\rho_j}$. Thus,

$$
\begin{aligned}
\sum_i \lambda_i + \sum_i \alpha_i + \quad & \sum_i \delta_i 1_{X_i>0} + \sum_i \mu_i 1_{X_i>0} + \sum_i \beta_i 1_{Y_i>0} + \sum_i \sum_j \gamma_{i,j} 1_{Y_i>0} + \sum_i \sum_{m=1}^{c_i-1} \rho_i \omega_i^m \mu_i 1_{Y_i>0} \\
+ \quad & \sum_i \sum_j \sum_{k\geq 1} \sum_{l=0}^{k-1} A_{i,j}^{(k)} \omega_i \omega_j^l \gamma_{i,j} 1_{Y_j>0} \\
= \quad & \sum_i \frac{\lambda_i}{\rho_i} 1_{X_i>0} \\
+ \quad & \sum_i \frac{\alpha_i}{\omega_i} 1_{Y_i>0} \\
+ \quad & \sum_i \beta_i \omega_i \\
+ \quad & \sum_i \rho_i \delta_i \\
+ \quad & \sum_i \sum_{m=0}^{c_i-1} \rho_i \omega_i^m \mu_i \\
+ \quad & \sum_i \sum_j \frac{\rho_i}{\rho_j} \omega_i^{c_i} \mu_i P(i,j) 1_{X_j>0} \\
+ \quad & \sum_i \sum_j \sum_{k\geq 1} \sum_{l=0}^{k-1} A_{i,j}^{(k)} \omega_i \omega_j^l \gamma_{i,j} \\
+ \quad & \sum_i \sum_j \sum_{k\geq 0} A_{i,j}^{(k)} \omega_i \omega_j^k \gamma_{i,j} 1_{X_i>0} \\
+ \quad & \sum_i \sum_j \sum_{m\geq 0} \sum_{k\geq 0} A_{i,j}^{(k)} \gamma_{i,j} \left(1 - \frac{\rho_i}{\rho_j}\right) \omega_i \omega_j^k 1_{X_i=m} 1_{X_j>0} \\
+ \quad & \sum_i \sum_j \sum_{m\geq 0} \sum_{k\geq 0} A_{i,j}^{(k)} \gamma_{i,j} \omega_i \omega_j^k 1_{X_i=m} 1_{X_j=0}
\end{aligned}
$$

Using that $\sum_{m\geq 0} 1_{X_i=m} = 1$, the last two terms of the above expressions simplify as follows:

$$
\begin{aligned}
\sum_i \lambda_i + \sum_i \alpha_i + \quad & \sum_i \delta_i 1_{X_i>0} + \sum_i \mu_i 1_{X_i>0} + \sum_i \beta_i 1_{Y_i>0} + \sum_i \sum_j \gamma_{i,j} 1_{Y_i>0} + \sum_i \sum_{m=1}^{c_i-1} \rho_i \omega_i^m \mu_i 1_{Y_i>0} \\
+ \quad & \sum_i \sum_j \sum_{k\geq 1} \sum_{l=0}^{k-1} A_{i,j}^{(k)} \omega_i \omega_j^l \gamma_{i,j} 1_{Y_j>0} \\
= \quad & \sum_i \frac{\lambda_i}{\rho_i} 1_{X_i>0} \\
+ \quad & \sum_i \frac{\alpha_i}{\omega_i} 1_{Y_i>0} \\
+ \quad & \sum_i \beta_i \omega_i \\
+ \quad & \sum_i \rho_i \delta_i \\
+ \quad & \sum_i \sum_{m=0}^{c_i-1} \rho_i \omega_i^m \mu_i \\
+ \quad & \sum_i \sum_j \frac{\rho_i}{\rho_j} \omega_i^{c_i} \mu_i P(i,j) 1_{X_j>0} \\
+ \quad & \sum_i \sum_j \sum_{k\geq 1} \sum_{l=0}^{k-1} A_{i,j}^{(k)} \omega_i \omega_j^l \gamma_{i,j} \\
+ \quad & \sum_i \sum_j \sum_{k\geq 0} A_{i,j}^{(k)} \omega_i \omega_j^k \gamma_{i,j} 1_{X_i>0} \\
+ \quad & \sum_i \sum_j \sum_{k\geq 0} A_{i,j}^{(k)} \gamma_{i,j} \left(1 - \frac{\rho_i}{\rho_j}\right) \omega_i \omega_j^k 1_{X_j>0} \\
+ \quad & \sum_i \sum_j \sum_{k\geq 0} A_{i,j}^{(k)} \gamma_{i,j} \omega_i \omega_j^k 1_{X_j=0}
\end{aligned}
$$

Using again that $1_{X_j=0} = 1 - 1_{X_j>0}$, we get that

$$
\begin{aligned}
\sum_i \lambda_i + \sum_i \alpha_i + \quad & \sum_i \delta_i 1_{X_i>0} + \sum_i \mu_i 1_{X_i>0} + \sum_i \beta_i 1_{Y_i>0} + \sum_i \sum_j \gamma_{i,j} 1_{Y_i>0} + \sum_i \sum_{m=1}^{c_i-1} \rho_i \omega_i^m \mu_i 1_{Y_i>0} \\
+ \quad & \sum_i \sum_j \sum_{k\geq 1} \sum_{l=0}^{k-1} A_{i,j}^{(k)} \omega_i \omega_j^l \gamma_{i,j} 1_{Y_j>0} \\
= \quad & \sum_i \frac{\lambda_i}{\rho_i} 1_{X_i>0} \\
+ \quad & \sum_i \frac{\alpha_i}{\omega_i} 1_{Y_i>0} \\
+ \quad & \sum_i \beta_i \omega_i \\
+ \quad & \sum_i \rho_i \delta_i \\
+ \quad & \sum_i \sum_{m=0}^{c_i-1} \rho_i \omega_i^m \mu_i \\
+ \quad & \sum_i \sum_j \frac{\rho_i}{\rho_j} \omega_i^{c_i} \mu_i P(i,j) 1_{X_j>0} \\
+ \quad & \sum_i \sum_j \sum_{k\geq 1} \sum_{l=0}^{k-1} A_{i,j}^{(k)} \omega_i \omega_j^l \gamma_{i,j} \\
+ \quad & \sum_i \sum_j \sum_{k\geq 0} A_{i,j}^{(k)} \omega_i \omega_j^k \gamma_{i,j} 1_{X_i>0} \\
+ \quad & \sum_i \sum_j \sum_{k\geq 0} A_{i,j}^{(k)} \gamma_{i,j} \left(1 - \frac{\rho_i}{\rho_j}\right) \omega_i \omega_j^k 1_{X_j>0} \\
+ \quad & \sum_i \sum_j \sum_{k\geq 0} A_{i,j}^{(k)} \gamma_{i,j} \omega_i \omega_j^k (1 - 1_{X_j>0})
\end{aligned}
$$

The above expression gets simplified as follows:

$$
\begin{aligned}
\sum_i \lambda_i + \sum_i \alpha_i + \quad & \sum_i \delta_i 1_{X_i>0} + \sum_i \mu_i 1_{X_i>0} + \sum_i \beta_i 1_{Y_i>0} + \sum_i \sum_j \gamma_{i,j} 1_{Y_i>0} + \sum_i \sum_{m=1}^{c_i-1} \rho_i \omega_i^m \mu_i 1_{Y_i>0} \\
+ \quad & \sum_i \sum_j \sum_{k\geq 1} \sum_{l=0}^{k-1} A_{i,j}^{(k)} \omega_i \omega_j^l \gamma_{i,j} 1_{Y_j>0} \\
= \quad & \sum_i \frac{\lambda_i}{\rho_i} 1_{X_i>0} \\
+ \quad & \sum_i \frac{\alpha_i}{\omega_i} 1_{Y_i>0} \\
+ \quad & \sum_i \beta_i \omega_i \\
+ \quad & \sum_i \rho_i \delta_i \\
+ \quad & \sum_i \sum_{m=0}^{c_i-1} \rho_i \omega_i^m \mu_i \\
+ \quad & \sum_i \sum_j \frac{\rho_i}{\rho_j} \omega_i^{c_i} \mu_i P(i,j) 1_{X_j>0} \\
+ \quad & \sum_i \sum_j \sum_{k\geq 1} \sum_{l=0}^{k-1} A_{i,j}^{(k)} \omega_i \omega_j^l \gamma_{i,j} \\
+ \quad & \sum_i \sum_j \sum_{k\geq 0} A_{i,j}^{(k)} \omega_i \omega_j^k \gamma_{i,j} 1_{X_i>0} \\
- \quad & \sum_i \sum_j \sum_{k\geq 0} A_{i,j}^{(k)} \gamma_{i,j} \frac{\rho_i}{\rho_j} \omega_i \omega_j^k 1_{X_j>0} \\
+ \quad & \sum_i \sum_j \sum_{k\geq 0} A_{i,j}^{(k)} \gamma_{i,j} \omega_i \omega_j^k
\end{aligned}
$$

We change the indices i and j of the sixth and the tenth term of the rhs and of the eighth term of the lhs, which gives

$$
\begin{aligned}
\sum_i \lambda_i + \sum_i \alpha_i + \quad & \sum_i \delta_i 1_{X_i>0} + \sum_i \mu_i 1_{X_i>0} + \sum_i \beta_i 1_{Y_i>0} + \sum_i \sum_j \gamma_{i,j} 1_{Y_i>0} + \sum_i \sum_{m=0}^{c_i-1} \rho_i \omega_i^m \mu_i 1_{Y_i>0} \\
+ \quad & \sum_i \sum_j \sum_{k\geq 1} \sum_{l=0}^{k-1} A_{j,i}^{(k)} \omega_j \omega_i^l \gamma_{j,i} 1_{Y_i>0} \\
= \quad & \sum_i \frac{\lambda_i}{\rho_i} 1_{X_i>0} \\
+ \quad & \sum_i \frac{\alpha_i}{\omega_i} 1_{Y_i>0} \\
+ \quad & \sum_i \beta_i \omega_i \\
+ \quad & \sum_i \rho_i \delta_i \\
+ \quad & \sum_i \sum_{m=0}^{c_i-1} \rho_i \omega_i^m \mu_i \\
+ \quad & \sum_i \sum_j \frac{\rho_j}{\rho_i} \omega_j^{c_j} \mu_j P(j,i) 1_{X_i>0} \\
+ \quad & \sum_i \sum_j \sum_{k\geq 1} \sum_{l=0}^{k-1} A_{i,j}^{(k)} \omega_i \omega_j^l \gamma_{i,j} \\
+ \quad & \sum_i \sum_j \sum_{k\geq 0} A_{i,j}^{(k)} \omega_i \omega_j^k \gamma_{i,j} 1_{X_i>0} \\
- \quad & \sum_i \sum_j \sum_{k\geq 0} A_{i,j}^{(k)} \gamma_{j,i} \frac{\rho_j}{\rho_i} \omega_j \omega_i^k 1_{X_i>0} \\
+ \quad & \sum_i \sum_j \sum_{k\geq 0} A_{i,j}^{(k)} \gamma_{i,j} \omega_i \omega_j^k
\end{aligned}
$$

We now check that equation with the indicator function $1_{X_i>0}$ is satisfied as well as the equation with indicator function $1_{Y_i>0}$ and the equation without indicator function (constant term).

- **Function** $1_{X_i>0}$:

$$\sum_i \delta_i + \sum_i \mu_i = \sum_i \frac{\lambda_i}{\rho_i} + \sum_i \sum_j \frac{\rho_j}{\rho_i} \omega_j^{c_j} \mu_j P(j,i) + \sum_i \sum_j \sum_{k\geq 0} A_{i,j}^{(k)} \omega_i \omega_j^k \gamma_{i,j}$$
$$- \sum_i \sum_j \sum_{k\geq 0} A_{j,i}^{(k)} \gamma_{j,i} \frac{\rho_j}{\rho_i} \omega_j \omega_i^k$$

The above equation holds by (FLOW-DP).

- **Function** $1_{Y_i>0}$:

$$\sum_i \beta_i + \sum_i \sum_j \gamma_{i,j} + \sum_i \sum_{m=0}^{c_i-1} \rho_i \omega_i^m \mu_i + \sum_i \sum_j \sum_{k\geq 1} \sum_{l=0}^{k-1} A_{i,j}^{(k)} \omega_j \omega_i^l \gamma_{j,i} = \sum_i \frac{\alpha_i}{\omega_i}$$

The above equation holds by (FLOW-EP).

- **Constant:**

$$\sum_i \lambda_i + \sum_i \alpha_i = \sum_i \beta_i \omega_i + \sum_i \rho_i \delta_i + \sum_i \sum_{m=0}^{c_i-1} \rho_i \omega_i^m \mu_i + \sum_i \sum_j \sum_{k\geq 1} \sum_{l=0}^{k-1} A_{i,j}^{(k)} \omega_i \omega_j^l \gamma_{i,j}$$
$$+ \sum_i \sum_j \sum_{k\geq 0} A_{i,j}^{(k)} \gamma_{i,j} \omega_i \omega_j^k$$

We now note that the last two terms of the rhs can be written in the following equivalent form:

$$\sum_i \lambda_i + \sum_i \alpha_i = \sum_i \beta_i \omega_i + \sum_i \rho_i \delta_i + \sum_i \sum_{m=0}^{c_i-1} \rho_i \omega_i^m \mu_i + \sum_i \sum_j \sum_{k\geq 0} \sum_{l=0}^{k} A_{i,j}^{(k)} \omega_i \omega_j^l \gamma_{i,j} \tag{B.1}$$

From (FLOW-EP), we have that

$$\sum_i \alpha_i = \sum_i \omega_i \beta_i + \sum_i \sum_j \omega_i \gamma_{i,j} + \sum_i \omega_i \sum_{m=0}^{c_i-1} \rho_i \omega_i^m \mu_i + \sum_i \sum_j \sum_{k\geq 1} \sum_{l=1}^{k} A_{i,j}^{(k)} \omega_j \omega_i^l \gamma_{j,i}$$

We replace this expression in (B.1) and we get

$$\sum_i \lambda_i + \sum_i \omega_i \beta_i + \sum_i \sum_j \omega_i \gamma_{i,j} + \sum_i \omega_i \sum_{m=0}^{c_i-1} \rho_i \omega_i^m \mu_i + \sum_i \sum_j \sum_{k\geq 1} \sum_{l=1}^{k} A_{i,j}^{(k)} \omega_j \omega_i^l \gamma_{j,i} =$$
$$\sum_i \beta_i \omega_i + \sum_i \rho_i \delta_i + \sum_i \sum_{m=0}^{c_i-1} \rho_i \omega_i^m \mu_i + \sum_i \sum_j \sum_{k\geq 0} \sum_{l=0}^{k} A_{i,j}^{(k)} \omega_i \omega_j^l \gamma_{i,j}$$

We simplify that above expression and it results

$$\sum_i \lambda_i + \sum_i \sum_j \omega_i \gamma_{i,j} + \sum_i \omega_i \sum_{m=0}^{c_i-1} \rho_i \omega_i^m \mu_i = \sum_i \rho_i \delta_i + \sum_i \sum_{m=0}^{c_i-1} \rho_i \omega_i^m \mu_i + \sum_i \sum_j \sum_{k\geq 0} A_{i,j}^{(k)} \omega_i \gamma_{i,j}$$

Using that $\sum_{k\geq 0} A_{i,j}^{(k)} = 1$ for all i, j, the above expression simplifies as follows:

$$\sum_i \lambda_i + \sum_i \sum_j \omega_i \gamma_{i,j} + \sum_i \omega_i \sum_{m=0}^{c_i-1} \rho_i \omega_i^m \mu_i = \sum_i \rho_i \delta_i + \sum_i \sum_{m=0}^{c_i-1} \rho_i \omega_i^m \mu_i + \sum_i \sum_j \omega_i \gamma_{i,j}$$

The term $\sum_i \sum_j \omega_i \gamma_{i,j}$ appears on both sides of the above expression and, therefore, they can be canceled:

$$\sum_i \lambda_i + \sum_i \omega_i \sum_{m=0}^{c_i-1} \rho_i \omega_i^m \mu_i = \sum_i \rho_i \delta_i + \sum_i \sum_{m=0}^{c_i-1} \rho_i \omega_i^m \mu_i. \tag{B.2}$$

From (FLOW-DP), we obtain the following expression:

$$\sum_i \rho_i \mu_i + \sum_i \rho_i \delta_i = \sum_i \lambda_i + \sum_i \sum_j \rho_u \mu_j \omega_j^{c_j} P(j,i) + \sum_i \sum_j \sum_{k \geq 0} A_{i,j}^k (\gamma_{i,j} \omega_j^k \omega_i \rho_i - \gamma_{j,i} \omega_i^k \omega_j \rho_j)$$

We now notice that $\sum_i \sum_j \gamma_{i,j} \omega_j^k \omega_i \rho_i = \sum_i \sum_j \gamma_{j,i} \omega_i^k \omega_j \rho_j$. Therefore, we above expression can be simplified as follows:

$$\sum_i \rho_i \mu_i + \sum_i \rho_i \delta_i = \sum_i \lambda_i + \sum_i \sum_j \rho_j \mu_j \omega_j^{c_j} P(j,i).$$

We now interchange the indices i and j of the last expression of the rhs and we get

$$\sum_i \lambda_i = \sum_i \rho_i \mu_i + \sum_i \rho_i \delta_i - \sum_i \sum_j \rho_i \mu_i \omega_i^{c_i} P(i,j).$$

We replace this expression in (B.2) and it results

$$\sum_i \rho_i \mu_i + \sum_i \rho_i \delta_i - \sum_i \sum_j \rho_i \mu_i \omega_i^{c_i} P(i,j) + \sum_i \omega_i \sum_{m=0}^{c_i-1} \rho_i \omega_i^m \mu_i = \sum_i \rho_i \delta_i + \sum_i \sum_{m=0}^{c_i-1} \rho_i \omega_i^m \mu_i.$$

Using that $\sum_i P(j,i) = 1$ and since $\sum_i \rho_i \delta_i$ appears on both sides, this expression gets simplified as follows:

$$\sum_i \rho_i \mu_i - \sum_i \rho_i \mu_i \omega_i^{c_i} + \sum_i \omega_i \sum_{m=0}^{c_i-1} \rho_i \omega_i^m \mu_i = \sum_i \sum_{m=0}^{c_i-1} \rho_i \omega_i^m \mu_i.$$

We move the negative terms to the rhs and we get

$$\sum_i \rho_i \mu_i + \sum_i \omega_i \sum_{m=0}^{c_i-1} \rho_i \omega_i^m \mu_i = \sum_i \sum_{m=0}^{c_i} \rho_i \omega_i^m \mu_i.$$

We now note that the lhs of the above expression satisfies that:

$$\sum_i \rho_i \mu_i + \sum_i \omega_i \sum_{m=0}^{c_i-1} \rho_i \omega_i^m \mu_i = \sum_i \rho_i \mu_i + \sum_i \sum_{m=1}^{c_i} \rho_i \omega_i^m \mu_i = \sum_i \sum_{m=0}^{c_i} \rho_i \omega_i^m \mu_i,$$

which implies that the desired result follows.

Appendix C. Proof of Lemma 2

- Consider an arbitrary (X, Y). As $\alpha_i > 0$ for all i, there exists sequence of EP arrivals with positive probability from $(\vec{0}, \vec{0})$ to $(0, Y')$ with $Y' \geq Y$ element-wise. Y' is larger than Y because we will eventually need some EP to move the fresh DP to their destination and finally reach any state (X, Y).

  Now, we have to prove that it is possible to make the DP arrive at any location. We proceed by induction cell after cell using the label ordering obtained in Lemma 1 in decreasing order. Let $I(i)$ be the label of cell number i. Let us consider an arbitrary cell j. The DP population one must reach is X_j. We have two cases:

– If $\lambda_j > 0$, we clearly have a sequence of X_j arrivals which leads to the destination. And these arrivals do not need energy. In this case $Y' = Y$. Due to the labeling in Lemma 1, these cells receive the smallest labels.

– If $\lambda_j = 0$. Let r the cell associated to the root of the tree which contains node j (with label $I(j)$). Due to the ordering, all the nodes among the path between r and j have labels k between $I(r)$ and $I(j)$ (i.e. $I(r) < k < I(j)$. We begin with the arrivals of X_j arrivals of fresh data packets at node r. We now prove by induction on the nodes among the path the following property: at each step it is possible to make all the data packets progress from a cell (say b) to next cell (say c) among the path from r to j, and, at the end of the step, the DP queue of cell b is empty. Again we have two cases,

  ∗ If the link between cell b and c is an arc of $\mathcal{R}$, we proceed by X_j services and routing and we have enough EP to perform these migrations as $Y' > Y$. Thus such actions have a positive probability.

  ∗ If the link between cell b and c is an arc of $\mathcal{G}$, one must perform a load balancing operation where cell c polls cell b. This is possible because the DP queue of cell c is empty by induction. Furthermore as the load balancing batch has a geometric distribution, moving a batch of exactly X_j DP has a positive probability.

  Thus by induction we have moved X_j DP from cell r to cell j. The induction relies to label $I()$ because load balancing operations only occur when the receiving DP queues are empty.

- We now have to exhibit a sequence of transitions between any (X, Y) and $(\vec{0}, \vec{0})$. We use label $J()$ to schedule the transitions. In the first step, all the queues which do not have enough energy to allow all the transitions during Step 2, increase their energy level (i.e. the number of EP). As $\alpha_i > 0$ for all i, all these events have a positive probability. Then in a second step, we empty all the DP queues. Finally, during a third step, If the number of EP is too large, at the end of the process, we will consider some events which make the energy goes to 0 in all the cells.

  For the second step, we proceed by induction on the cells in the orders of $J()$ labels to establish that all DP queues with labels $J()$ smaller than i may be emptied by a sequence of events with positive probability.

  – The smallest (with this label) cells are the sinks. As already mentioned, a DP queue in a sink can be emptied either due to a routing out of the network (with rate δ_i) or a movement which fails due to the lack of energy. And the energy is missing here because it has previously leaked. Therefore we have a sequence of events with a positive probability which leads to empty DP queues in all the sinks.

  – Now consider cell i with label $J(i)$. By construction, cell i is not a sink. Due to the property of this label, all the DP queues among the path from i to the root of its tree r are decreasing.

Therefore, by induction their DP queues are empty. To move X_i data packets from node i to node r, we proceed by another induction among the path as in the previous case. At the current step of this induction, we have X_i packets at DP queue b and we have to send them to the next cell (say c). If the link between cell b and c is an arc of $\mathcal{R}$, we proceed by X_j services and routing and we have enough EP to perform these migrations as $Y' > Y$. Thus such actions have a positive probability. And if the link between cell b and c is an arc of $\mathcal{G}$, we perform a load balancing operation where cell c polls cell b. This is possible because the DP queue of cell c is empty by induction. Again, the load balancing batch has a geometric distribution, thus moving a batch of exactly X_j DP has a positive probability. At the end of this sequence of events, X_i have have reached the sink associated with node i. And due to the properties of the cell they can leave the network.

Let us now describe the third step which is used to delete the remaining EP. As $\beta_i + \sum_j \gamma_{i,j} > 0$ there exist a sequence of transitions which empties all the EP queues. Indeed, either a leak or a failed load balancing operation lead to a loss of energy packets. Finally we have a sequence of transitions from (X, Y) to $(\vec{0}, \vec{0})$.

□

Appendix D. Example of an irreducible network that is not open

We consider a network formed by two cells a and b. The parameters are set to the following ones: $\lambda_a = 1, \lambda_b = 2, \alpha_a = 2, \alpha_b = 1, \beta_a = 0, \beta_b = 0, \mu_a = 0, \mu_b = 0, \delta_a = 4, \delta_b = 4, \gamma_{a,b} = 4, \gamma_{b,a} = 0$. From these parameters, we notice that the load balancing takes places to move DPs from b to a. We consider that $A^{(k)}_{a,b} = 1$ for $k = 1$. We also note that, for these parameters, the first three conditions of open network are satisfied, but the last one does not hold for cell b. Therefore, the network is not open and we cannot apply Lemma 2 .

We now show that the Markov chain associated with this network is irreducible. Indeed, as the arrival rates are all positive, it is possible to reach any state (X_a, Y_a, X_b, Y_b) from the empty state $(0, 0, 0, 0)$. And, the DP queues may be emptied with departures with rate δ_i while the EP queues may be emptied by the polling operations of cell b by cell a which use a positive number of EP on both EP queues. Therefore, there also exists a sequence of transitions with positive probabilities from any state (X_a, Y_a, X_b, Y_b) to state $(0, 0, 0, 0)$.

Taking into account the zero values of the parameters and that the graph $\mathcal{H}$ is a DAG, the flow equations of the EPs are simplified as follows:

$$\omega_a = \frac{\alpha_a}{\gamma_{a,b}}, \omega_b = \frac{\alpha_b}{\omega_a \gamma_{a,b}},$$

whereas the flow equation of the DPs:

$$\rho_a = \frac{\lambda_a + \gamma_{a,b}\omega_a\omega_b\rho_a}{\delta_a}, \rho_b = \frac{\lambda_b - \gamma_{a,b}\omega_b\omega_a\rho_a}{\delta_b}.$$

We solve numerically this system of equation for the values of the parameters we have defined above and we obtain that

$$\omega_a = 1/2, \omega_b = 1/2, \rho_a = 1/3, \rho_b = 5/12.$$

Therefore, the chain is irreducible and the solution of the flow equation exists. As a result, one can apply Theorem 2 even if Lemma 2 does not hold.

Appendix E. Proof of Theorem 3

We prove that, under the conditions of Definition 3, the assumptions of Brouwer's theorem hold [22], i.e, there exists $\mathcal{D}$ which is a non empty compact subset of $\mathbb{R}_+^{2N}$ such that the flow equation function maps $\mathcal{D}$ onto itself, and as a result, a fixed point exists in $\mathcal{D}$ for the flow equations.

Let $F(\omega, \rho)$ be the vector of size N whose component i is $F_i(\omega, \rho)$. Likewise, we define $G(\omega, \rho)$ as the vector of size N whose component i is $G_i(\omega, \rho)$. First, we remark that as $\beta_i + \sum_j \gamma_{i,j} > 0$ and $\mu_i + \delta_i > 0$, the functions $F_i()$ and $G_i()$ are all continuous on $\mathbb{R}_+^{2N}$. We now aim to find $\mathcal{D}$, which is a compact subset of $\mathbb{R}_+^{2N}$ with a non empty interior such that $(F(\mathcal{D}), G(\mathcal{D})) \subset \mathcal{D}$. Remember that compact subsets of $\mathbb{R}_+^{2N}$ are bounded and closed subsets. We first study the constraints on $\mathcal{D}$ associated with $F_i()$ for all i and then the ones associated with $G_i()$.

- **Function F_i:**

  We first observe that

  $$F_i(\omega, \rho) = \frac{\alpha_i}{\beta_i + \mu_i\rho_i \sum_{m=0}^{c_i-1}\omega_i^m + \sum_j \gamma_{i,j} + \sum_j \sum_{k\geq 1}\sum_{l=0}^{k-1} A_{j,i}^{(k)}\omega_j(\omega_i)^l\gamma_{j,i}} \leq \frac{\alpha_i}{\beta_i + \sum_k \gamma_{i,j}}.$$

  Now, using condition (HYP1), we conclude that $F_i(\omega, \rho) \leq 1$. Therefore, one can choose the subset [0, 1] for the first N elements of the set $\mathcal{D}$, i.e., for the elements associated with function F_i .

- **Function G_i:**

  Note that we must establish that functions $G_i(\omega, \rho)$ are lower and upper bounded due to the substraction operation in the denominator. Let us consider first the upper bound. We see that

  $$\begin{aligned} G_i(\omega, \rho) &= \frac{\lambda_i + \sum_j \mu_j\rho_j\omega_j^{c_j}P(j,i) + \sum_j\sum_{k\geq 0}(A_{i,j}^{(k)}\gamma_{i,j}\omega_i\omega_j^k\rho_i - A_{j,i}^{(k)}\gamma_{j,i}\omega_j\omega_i^k\rho_j)}{\mu_i + \delta_i} \\ &\leq \frac{\lambda_i + \sum_j \mu_j\rho_j\omega_j^{c_j}P(j,i) + \sum_j\sum_{k\geq 0}A_{i,j}^{(k)}\gamma_{i,j}\rho_i\omega_i\omega_j^k}{\delta_i + \mu_i}. \end{aligned}$$

As $\omega_j < 1$ for all j, it results:

$$\begin{aligned}\frac{\lambda_i + \sum_j \mu_j \rho_j \omega_j^{c_j} P(j,i) + \sum_j \sum_{k\geq 0} A_{i,j}^{(k)} \gamma_{i,j} \rho_i \omega_i \omega_j^k}{\delta_i + \mu_i} &< \frac{\lambda_i + \sum_j \mu_j \rho_j P(j,i) + \sum_j \sum_{k\geq 0} A_{i,j}^{k} \gamma_{i,j} \rho_i}{\delta_i + \mu_i} \\ &= \frac{\lambda_i + \sum_j \mu_j \rho_j P(j,i) + \sum_j \gamma_{i,j} \rho_i \sum_{k\geq 0} A_{i,j}^{(k)}}{\delta_i + \mu_i} \\ &= \frac{\lambda_i + \sum_j \mu_j \rho_j P(j,i) + \sum_j \gamma_{i,j} \rho_i}{\delta_i + \mu_i}.\end{aligned}$$

Let $H_i(\rho) = \frac{\lambda_i + \sum_j \mu_j \rho_j P(j,i) + \sum_j \gamma_{i,j} \rho_i}{\delta_i + \mu_i}$. Thus, from the above reasoning, we have that $G_i(\omega, \rho) < H_i(\rho)$ for all i. Let us now consider the Jackson network with the parameters given in (HYP3) of Definition 3. The solution of the flow equation for this Jackson network is:

$$\mu_i \rho_i^* = (\lambda_i + \sum_j \gamma_{i,j}) \frac{\mu_i}{\mu_i + \delta_i} + \sum_j \mu_j \rho_j^* \left[\frac{\mu_i P(j,i)}{\mu_i + \delta_i} \right].$$

Rearranging both sides of the expression, we get after simplification:

$$\rho_i^* = \frac{\lambda_i + \sum_j \mu_j \rho_j^* P(j,i) + \sum_j \gamma_{i,j}}{\mu_i + \delta_i}.$$

As $\rho_i^* < 1$, and $\gamma_{i,j} \geq 0$ for all i and j, we have for all $\rho_i \leq \rho_i^* < 1$:

$$\rho_i^* \geq \frac{\lambda_i + \sum_j \mu_j \rho_j^* P(j,i) + \sum_j \gamma_{i,j} \rho_i}{\mu_i + \delta_i}.$$

Now, we remark that the functions $H_i()$ are all non-decreasing functions. Therefore if $\rho_i < \rho_i^*$, we have from the above reasoning that:

$$G_i(\omega, \rho) < H_i(\rho) \leq \rho_i^*.$$

Let us consider now the lower bound. As $0 \geq \omega_i < 1$ for all i, we have:

$$\begin{aligned}G_i(\omega, \rho) &= \frac{\lambda_i + \sum_j \mu_j \rho_j \omega_j^{c_j} P(j,i) + \sum_j \sum_{k\geq 0} (A_{i,j}^{(k)} \gamma_{i,j} \omega_i \omega_j^k \rho_i - A_{j,i}^{(k)} \gamma_{j,i} \omega_j \omega_i^k \rho_j)}{\mu_i + \delta_i} \\ &\geq \frac{\lambda_i - \sum_j \sum_{k\geq 0} A_{j,i}^{(k)} \gamma_{j,i} \rho_j}{\delta_i + \mu_i} = \frac{\lambda_i - \sum_j \gamma_{j,i} \rho_j}{\delta_i + \mu_i}.\end{aligned}$$

Thus if $\rho_i \leq \rho_i^* < 1$, we have

$$G_i(\omega, \rho) \geq \frac{\lambda_i - \sum_j \gamma_{j,i} \rho_j^*}{\delta_i + \mu_i} > \frac{\lambda_i - \sum_j \gamma_{j,i}}{\delta_i + \mu_i}$$

Therefore due to (HYP2), $G_i(\omega, \rho) \geq 0$, if $\rho_i \leq \rho_i^*$ for all i.

Thus we set $\mathcal{D} = [0,1]^N \times [0, \rho_1^*] \times [0, \rho_2^*] \times \cdots \times [0, \rho_N^*]$, and the assumptions of Brouwer's theorem hold.

The proof is complete: a fixed point for the flow equation exists in $\mathcal{D}$.