HAL
open science

Activation and consensus control of a three-node server network cluster via hybrid approach

Carolina Albea-Sanchez, Alexandre Seuret, Luca Zaccarian

To cite this version:

Carolina Albea-Sanchez, Alexandre Seuret, Luca Zaccarian. Activation and consensus control of a three-node server network cluster via hybrid approach. Nonlinear Analysis: Hybrid Systems, 2016, 22, pp.16-30. hal-01278146

HAL Id: hal-01278146

https://hal.science/hal-01278146v1

Submitted on 23 Feb 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Activation and consensus control of a three-node server network cluster via hybrid approach

Carolina Albea[a,b,*], Alexandre Seuret[a,c], Luca Zaccarian[a,c,d]

[a] *CNRS, LAAS, 7 avenue du Colonel Roche, 31077 Toulouse, France*
[b] *Univ. de Toulouse, UPS, LAAS, F-31400, Toulouse, France.*
[c] *Univ. de Toulouse, LAAS, F-31400 Toulouse, France*
[d] *Dipartimento di Ingegneria Industriale, University of Trento, Italy*

Abstract

We consider the management problem for a cluster of three identical servers treating an external service request. We assume that each server has a constant computational speed and we build a deterministic hybrid dynamical model of the overall system. Then, inspired by consensus theory, we propose a centralized control law distributing the computational load among the servers and deciding when to switch on or off each server, for energy efficiency, while ensuring that the queue of each server asymptotically converges to a desirable level. We formally prove the properties of the proposed law and highlight some interesting peculiarities in terms of non-uniform convergence. While we prove our results for the case of a constant request rate, we show the effectiveness of the proposed strategies by simulations in the more general cases of a time-varying request rate.

Keywords: Consensus, activation control, hybrid dynamical system

1. Introduction

Distribution of computational load across available resources is referred to as the "load balancing problem" in the literature. This may refer to the classical problem of distributing computational load over a network of servers having a parallel computer architecture [20], or balancing energy delivery network architectures for the decentralized hierarchical integration of microgrids [1], for instance. Various taxonomies of load balancing algorithms exist. The proposed solutions arise from deterministic and gradient-based methods, stochastic, or optimization-based. A comparison of several deterministic methods is provided in [21].
Since the load balancing problem is of great relevance from a technological viewpoint, the algorithmic literature in this context is extremely vast and will not be surveyed here (the reader is referred, e.g., to [8, 13]

*Corresponding author C. Albea. Tel. +33-561337815. Fax +33-561336411.
Email addresses: calbea@laas.fr (Carolina Albea), aseuret@laas.fr (Alexandre Seuret), zaccarian@laas.fr (Luca Zaccarian)

and references therein). The approach taken in this paper is that of a control viewpoint on this problem, which is seldom found in the existing literature. Notable exceptions are the works in [2, 12], where load balancing is carried out in a static context given a certain communication network architecture, in addition to the works cited in the recent overview [19], where load balancing is one of the many mentioned challenging problems. The results summarized in [19] are mainly inspired by the works in [20, 19], where the focus is mainly on the effect of delays on the stability properties induced by the control law (see also the follow-up work in [11]). Some recent work employing feedback control strategies to provide reduced service times when running programs in "the cloud", or equivalently in a big network cluster, has been published in [3]. While different in nature, our work is similar in spirit to this last one as it provides a different system theoretic solution to the problem of controlling a computer network activity.

Beyond load balancing, this paper also addresses activity management for a network cluster, wherein the control algorithm is also responsible for deciding what servers are active/inactive at a specific time. This control decision affects the network architecture that is not fixed a priori but time-varying and dependent on the adopted control strategy. One of the difficulties behind the activity management problem relies on the relevant degrees of freedom in the control design of switching on or off servers (or nodes) in the cluster. The objective is to perform a desirable choice of the active nodes and possibly to increase the performance in terms of a reduction of the power consumption. While from the computer science point of view the systematic decision about on-off switching of different nodes has been proposed since a long time (see, for example, for the case of a sensor network, the works [14, 22]), this task appears to be more challenging in a dynamical systems framework due to the intrinsic hybrid nature of the dynamics where switching on and off different nodes induces discontinuities on the model behavior. Nevertheless, modeling a load balancing algorithm comprising on-off switching rules may allow to prove desirable stability and/or convergence properties of the whole network cluster from a system theoretic viewpoint.

This is the approach taken in this paper, where the dynamics of a network cluster is described as a hybrid dynamical system whose solutions may continuously flow according to some differential equation (thereby describing the continuous evolution of the network) and may instantaneously jump according to some re-initialization rule that would comprise, for example, switching on or off a specific node. An example of switched multi-agent systems can be found, for instance, in [7, 16], where the switching signals are seen as external disturbances affecting the graph topology, and the associated analysis aims at proving robustness of the control algorithms with respect to these graph topology variations. Likewise, an example of consensus algorithm applying hybrid dynamic system is proposed in [5]. In this case, there is not control in the graph topology. Here we adopt a self-adjusting multi-agent control strategy and we develop our results using the notation and the framework proposed in the recent work [10, 9]. The control law that we propose

follows a novel paradigm for controlling the cluster activity, where enabling or disabling the cluster nodes is triggered by a hybrid control law governing both the continuous and the discrete dynamics of the cluster. In particular, we develop our results by combining the theory of multi-agent systems (see, e.g., [15]), and the above mentioned theory of hybrid systems [10]. As a first pioneering result in this new research direction of self-adjusting multi-agent systems, we focus on a simple cluster comprising three nodes and we assume that a few scalar quantities related to the network activity are available to all nodes. Within this context, the hybrid controller that we propose provides very desirable transients and desirable steady-state properties in terms of guaranteed convergence to some reasonable operating conditions of the cluster. One interesting property that emerges from our analysis is that this convergence is not uniform, thereby revealing (also in light of the fundamental results in [10]) that the set where trajectories converge is attractive but not stable. This somewhat peculiar feature is interpreted and explained in the paper. Note that existing works address consensus problems using hybrid techniques (see, e.g., [6, 5, 4] and references therein), nevertheless, our work is the first one where mode changes (namely activation/deactivation) is cast as a feedback decision in a hybrid dynamical systems formalism.

The paper is organized as follows. In Section 2 we formulate our problem and a few preliminary results on the proposed hybrid model are established. Then we describe the proposed control law in Section 3, where we state our main convergence result and provide some discussions. The proof of the main theorem is given in Section 4, where it is broken into a number of lemmas clarifying the main features of the proposed algorithm. Simulation results are finally reported in Section 5, to illustrate from a practical viewpoint the features of the proposed control law. Conclusions and future directions are then drawn in Section 6.

Notation: The set of non-negative reals is denoted by $\mathbb{R}_{\geq 0}$. Given a vector w, $|w|$ denotes its Euclidean norm. The notation $\mathbf{1}$ represents the vector of appropriate dimensions whose entries are all equal to one. Given a set $\mathcal{D}$, $\bar{\mathcal{D}}$ denotes its closure and $\mathcal{D}^c$ denotes its complement.

2. Problem formulation

2.1. Source and nodes

Consider a source $\mathcal{S}$ that distributes suitable service request rate $w \in \mathbb{R}_{>0}$ among a set of three servers (or nodes) $\mathcal{N} := \{1, 2, 3\}$. The objective of the cluster is to perform suitable processing of this computational load, while operating the servers (nodes) at their maximum computational capacity, to maximize efficiency. We represent the information flow through the network via the service requests coming from the source and the processing service provided by the nodes. The nodes are assumed to have limited and fixed service rate $y \in \mathbb{R}_{>0}$, denoted as *computational capacity* (see Figure 1).

The core contribution of this paper within the considered scenario is to propose a feedback control scheme

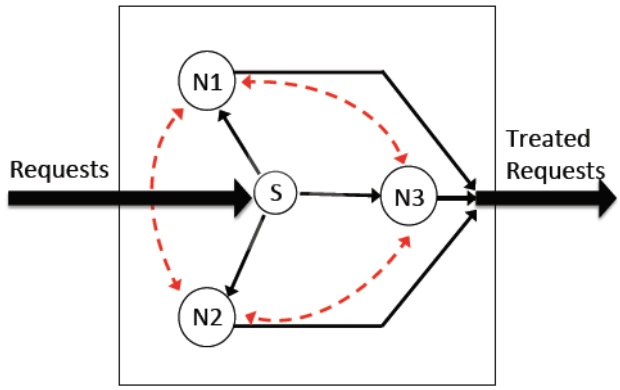


Figure 1: The considered three-node cluster.

for suitably switching on and off each node, which adjusts the right number of nodes to be activated in order to process the amount of requests from the source. For this reason, the following assumption is introduced

Assumption 1. *The total processing rate $w \in \mathbb{R}_{>0}$ requested by the source is constant and the service rate $y \in \mathbb{R}$ provided by each active server satisfies*

$$w = yn^* = 2y. \tag{1}$$

Assumption 1 means that there exists a number $n^* = 2$ of active nodes in the network composed of three nodes ($n = 3$), that can treat exactly the number of requests provided by the source, while operating at the fixed service rate y.

Remark 1. Assumption 1 is restrictive but represents the first and simplest context that we can characterize with the novel scheme proposed in this paper. While we establish formal results about the scheme for this very peculiar situation, we will carry out simulations for a more general situation where the number of servers in the cluster is much larger and the requested computational rate w is replaced by a time-varying function $t \mapsto w(t)$. See Figure 3 and the corresponding descriptions in Section 5. ⌟

2.2. Dynamics of the nodes: a saturated integrator model

Associate with each node $i \in \mathcal{N}$ a state variable $\alpha_i \in \{0, 1\}$ whose value indicates whether the node should be in a quiescent state ($\alpha_i = 0$), typically associated with low consumption, or in an active state ($\alpha_i = 1$) that, as clarified later, corresponds to full service rate. We say that a node $i \in \mathcal{N}$ is *active* if $\alpha_i = 1$ and it is *inactive* if $\alpha_i = 0$. For notational simplicity we introduce the following *network activity vector* $\alpha \in \{0, 1\} \times \{0, 1\} \times \{0, 1\}$:

$$\alpha = [\alpha_1 \quad \alpha_2 \quad \alpha_3]^T,$$

which is a (discontinuously) time-varying quantity. To perform suitable processing of the workload coming from the source, we equip each node with a queue which is able to locally store the requests arriving from the source and which are still not processed. To this aim, we introduce for each $i \in \mathcal{N}$ the state *queue length* $q_i \in \mathbb{R}_{\geq 0}$, whose variation $\dot{q}_i$ corresponds to the sum of the requested rate coming from the source (w/n, where n is the number of active nodes, following the notation given in Assumption 1), the control input, u_i, and the maximum processing rate y. If the queue q_i is strictly positive, the dynamics of the queues can be summarized as follows

$$\dot{q}_i = \alpha_i(u_i + w/n) - y. \tag{2}$$

In this paper, we design two control actions. The first one u_i refers to the continuous control action relevant during continuous evolution (flowing), which represents the flux exchanged among the nodes and, thus, should induce consensus among the queue lengths of the active nodes. The second control action α_i is binary (on/off) and corresponds to a suitable feedback selection of the activation and deactivation of each server. Model (2) shows that, if a node is active ($\alpha_i = 1$) and the queue length q_i is strictly positive, the dynamics of the queue is affected by the control input u_i and the request rate w/n. On the other hand, if the node is inactive, the queue length is strictly decreasing at its maximum rate $-y$, until it reaches zero. An additional logic state $\beta_i \in \{0, 1\}$ is then introduced, which forces the right hand side of the flow equation to zero once the queue becomes empty, in a similar way to the behavior of thyristors, when their current drops down to zero. The logic state β_i also ensures that the service requests still present in the queue are suitably processed after the node is deactivated (namely after α_i switches to zero). The overall hybrid dynamics of each node then comprises an internal state $\left[\begin{smallmatrix} q_i \\ \beta_i \end{smallmatrix}\right]$ and the action of the external input $\left[\begin{smallmatrix} u_i \\ \alpha_i \end{smallmatrix}\right]$ designed in Section 3. Using the notation $x_i = [q_i \ \beta_i \ \alpha_i]^T \in \mathbb{R}_{\geq 0} \times \{0, 1\}^2$ and recalling that the constant scalars w and y represent the requested processing rate and the service rate of each node, respectively, each node dynamics is written as:

$$\begin{cases} \dot{q}_i &= \beta_i \left(\alpha_i(u_i + w/n) - y\right), & x_i \in \mathcal{C}_i^0, \\ \dot{\beta}_i &= 0, & \\ q_i^+ &= 0, & x_i \in \mathcal{D}_i^0, \\ \beta_i^+ &= \alpha_i, & \end{cases} \tag{3a}$$

where, for each $i \in \mathcal{N}$, the flow and jump sets are given by:

$$\begin{aligned} \mathcal{C}_i^0 &= ([0, +\infty) \times \{0, 1\} \times \{0, 1\}) \setminus ([0, +\infty] \times \{0\} \times \{1\}), \\ \mathcal{D}_i^0 &= D_i^{\text{act}} \cup D_i^{\text{empty}}, \\ D_i^{\text{act}} &= \{x_i : \beta_i = 0 \text{ and } \alpha_i = 1\} \\ D_i^{\text{empty}} &= \{x_i : q_i = 0, \beta_i = 1, \text{ and } \alpha_i = 0\}. \end{aligned} \tag{3b}$$

In the flow dynamics (3a), the term w/n corresponds to sharing the requests rate w coming from the

source among the total number $n := \alpha^T \mathbf{1}$ of active nodes, which during flowing can be expressed as:

$$n = \alpha^T \mathbf{1} = \alpha^T \alpha = \alpha^T \beta. \tag{4}$$

Indeed, from the definition of $\mathcal{C}_i^0$, flowing is not possible if $\beta_i = 0$ and $\alpha_i = 1$. Note that for simplicity the total rate is split in equal parts among the active nodes.

A different selection could be compensated by suitable choices of the control input u_i, which is part of our control design.

2.3. Connection graph among nodes

According to the activities of the nodes $\alpha = [\alpha_1, \alpha_2, \alpha_3]^T$, we classically define the adjacency matrix $A(\alpha) = [a_{ih}(\alpha)]$, where

$$a_{ih}(\alpha) = \begin{cases} \alpha_i \alpha_h & \text{if} \quad i \neq h, \\ 0 & \text{if} \quad i = h \end{cases} \qquad \forall\, i, h \in \mathcal{N}. \tag{5}$$

and the diagonal matrix $\Delta(\alpha) = [\delta_{ih}(\alpha)]$, where

$$\delta_{ih}(\alpha) = \begin{cases} 0 & \text{if} \quad i \neq h, \\ \sum_{k \in \mathcal{N}, k \neq i} a_{ik}(\alpha) & \text{if} \quad i = h \end{cases} \qquad \forall\, i, h \in \mathcal{N}.$$

Therefore, the Laplacian matrix representing the undirected graph of interconnections within the cluster is given by

$$L(\alpha) = \Delta(\alpha) - A(\alpha). \tag{6}$$

The Laplacian of an undirected graph is a symmetric positive semi-definite matrix. It is worth mentioning that the resulting Laplacian $L(\alpha)$ represents the complete graph among the nodes that are on (comprising the red arrows in Figure 1 not insisting on inactive nodes). Matrix $L(\alpha)$ satisfies the following useful property, whose proof follows from standard properties of Laplacian matrices and from the construction of $L(\alpha)$ (see [15] for details on the construction of the Laplacian matrix).

Property 1. *The following relations hold*

$$L(\alpha)\mathbf{1} = L(\alpha)\alpha = L(\alpha)\beta = 0, \quad \mathbf{1}^T L(\alpha) = \alpha^T L(\alpha) = \beta^T L(\alpha) = 0. \tag{7}$$

2.4. Control objectives

The goal of this paper is to propose a suitable control law that both coordinates the activation/deactivation of the nodes and assigns the inputs u_i to each node to achieve the following goals along the solutions to the proposed closed loop (note that solutions are parametrized by both t and j but their domain is guaranteed to be unbounded in the ordinary time direction, thereby making the limits below well posed):

- The number of active nodes satisfies,

$$\lim_{t\to+\infty}(n(t,j)-n^*)=0. \tag{8}$$

where n^* is defined in (1).

- Given a desired queue length $q^*>0$ and a desired range $\varepsilon>0$, the queue length q_i, for any $i=1,2,3$, must satisfy

$$\lim_{t\to+\infty}\alpha_i(t,j)=1 \;\Rightarrow\; \lim_{t\to+\infty}q_i(t,j)\in\left[q^*-\frac{\varepsilon}{2},q^*+\frac{\varepsilon}{2}\right], \tag{9}$$

$$\lim_{t\to+\infty}\alpha_i(t,j)=0 \;\Rightarrow\; \lim_{t\to+\infty}q_i(t,j)=0. \tag{10}$$

- All the requested processing is treated during the transient:

$$\sum_{i\in\mathcal{N}}\dot{q}_i=w-\sum_{i\in\mathcal{N}}\beta_i y,\quad \forall t\geq 0. \tag{11}$$

Remark 2. Note that the first two control objectives above essentially correspond to a practical load balancing requirement among the nodes that will eventually remain active, together with the requirement that the nodes eventually remaining inactive converge to having an empty queue (so that they can be switched off). Condition (8) can be enforced in the simplified setting of this paper because of Assumption 1. However, in a more realistic context, such as the one simulated in Figure 3, Assumption 1 will not hold and a time-varying processing load $t\mapsto w(t)$ will appear, so that the system may keep activating and deactivating nodes. Indeed, even with a constant w different from a multiple of y, one may have persistent activation/deactivation of nodes because we force all enabled servers to always work at their maximum rate y.

In this case, condition (8) will not hold, even though conditions (9)-(11) may still be true. This is illustrated, for example, in Figure 3d where one clearly sees that $n(t,j)$ does not converge (lower plot), so (8) does not hold, but the queues of nodes 1, 2, 3 and 14, which are eventually active, reach and maintain consensus.

The third requirement is a flow constraint on the solutions comprising the fact that the sum of the instantaneous difference between the total requested service rate (w) and the actual total service rate of the nodes ($\sum_{i\in\mathcal{N}}\beta_i y$) is losslessly stored within the queues of the nodes ($\sum_{i\in\mathcal{N}}q_i$). This requirement can be satisfied by a suitable selection of the control law. ⌟

We may well represent the two first control objectives in terms of an error variable e, which is defined by

$$e=\begin{bmatrix}\alpha^T\tilde{q}\\ \tilde{n}\end{bmatrix}, \tag{12a}$$

$$\text{where } \tilde{n} = n - n^*, \text{ and } \tilde{q} = \begin{bmatrix} q_1 - q^* \\ q_2 - q^* \\ q_3 - q^* \end{bmatrix}. \tag{12b}$$

In the next section, we will develop a control scheme that fulfills these control objectives based on a consensus algorithm and a hybrid modeling strategy.

3. Proposed control law

3.1. Load balancing control law

Using model (3) and the interconnection (Laplacian) matrix (6), we may select a dynamic distributed controller, that represents the flux exchanged among the nodes, corresponding to

$$u_i = k_p \sum_{h \in \mathcal{N}} a_{ih}(\alpha)(q_h - q_i) = \alpha_i k_p \sum_{h \in \mathcal{N}} \alpha_h (q_h - q_i), \quad \forall i = 1, 2, 3. \tag{13}$$

The coefficient k_p is a strictly positive parameter and coefficients a_{ih} are defined in (5). Control law (13) is a classical consensus algorithm for simple integrator multi-agent systems [15, 17], which ensures that the lengths of the queues converge to an agreement. This control law is easily implemented by sharing the queue length information among neighboring nodes.

Gathering all the dynamics of the queues in (3), the following equations are derived

$$\dot{q} \;=\; u + \alpha \frac{w}{n} - \beta y, \qquad u \;=\; -k_p L(\alpha) q, \tag{14}$$

where $q = \left[\begin{smallmatrix} q_1 \\ q_2 \\ q_3 \end{smallmatrix}\right]$, $u = \left[\begin{smallmatrix} u_1 \\ u_2 \\ u_3 \end{smallmatrix}\right]$ and where $L(\alpha)$ is the Laplacian matrix defined in (6). Equations (14) have been obtained from the entries of $L(\alpha)$, which ensure that the equality $\alpha_i u_i = u_i$ holds for any $i = 1, 2, 3$, and thanks to the fact that, according to dynamics (3), no flowing is possible for solutions in the case when $\alpha_i = 1$ and $\beta_i = 0$, so that the term β_i multiplying u_i and w in (3a) can be dropped.

In the following lemma we establish a useful property of the error term $\alpha^T \tilde{q}$, exploited in the proof of our main result.

Lemma 1. *The quantity $\alpha^T \tilde{q}$ represents the sum of the active queues and satisfies*

$$\alpha^T \dot{\tilde{q}} = -\tilde{n} y. \tag{15}$$

Proof. From equation (14), we get along flow:

$$\alpha^T \dot{\tilde{q}} = -k_p \left(\alpha^T L(\alpha)\right) (\tilde{q} + q^* \mathbf{1}) + (\alpha^T \alpha / n) w - \alpha^T \beta y.$$

We use $\alpha^T L(\alpha) = 0$ from Property 1 together with the definitions of n in (4) and n^* in (1) to show that $\alpha^T \dot{\tilde{q}} = (n^* - n)y = -\tilde{n}y$, which establishes (15). □

The next proposition establishes one of the control objectives specified in Section 2.4.

Proposition 1. *The distributed control law* (13) *satisfies the complete processing condition* (11).

Proof. Consider equations (14) and take the sum over all the nodes, to get

$$\sum_{i\in\mathcal{N}} \dot{q}_i = \sum_{i\in\mathcal{N}} u_i + \underbrace{\alpha^T \mathbf{1}\frac{w}{n}}_{=w} - \sum_{i\in\mathcal{N}} \beta_i y,$$

therefore the result follows if:

$$\sum_{i\in\mathcal{N}} u_i = \alpha^T u = -k_p \left(\alpha^T L(\alpha)\right) q = 0, \tag{16}$$

which follows from the relation $\alpha^T L(\alpha) = 0$, established in Property 1. □

3.2. Hybrid control scheme

This section is dedicated to the introduction of an appropriate model to represent the continuous and discrete-time behavior of the system. In particular, the continuous motion comprises (14) while the discrete one is the novelty of this paper and comprises the feedback decisions about switching on/off the nodes.

Hence, we propose a hybrid formulation that gathers the continuous and discrete dynamics in a single model:

$$\begin{array}{rclcrcl} \dot{x}_i & = & f_i(x_i, u_i, w), & & (x_i, e) & \in & \mathcal{C}_i, \\ x_i^+ & \in & G_i(x_i), & & (x_i, e) & \in & \mathcal{D}_i, \end{array} \tag{17a}$$

for $i = 1, 2, 3$, where [1] for each $i \in \{1, 2, 3\}$,

$$G_i(x_i, e) := \begin{cases} \{g_i^{\text{on}}(x_i)\}, & \text{if } (x_i, e) \in \mathcal{D}_i^{\text{on}}, \\ \{g_i^{\text{off}}(x_i)\}, & \text{if } (x_i, e) \in \mathcal{D}_i^{\text{off}} \backslash \tilde{\mathcal{D}}_i^0, \\ \{g_i^0(x_i)\}, & \text{if } (x_i, e) \in \tilde{\mathcal{D}}_i^0 \backslash \mathcal{D}_i^{\text{off}}, \\ \{g_i^0(x_i)\} \cup \{g_i^{\text{off}}(x_i)\}, & \text{if } (x_i, e) \in \mathcal{D}_i^{\text{off}} \cap \tilde{\mathcal{D}}_i^0, \end{cases} \tag{17b}$$

$$x_i := \begin{bmatrix} q_i \\ \beta_i \\ \alpha_i \end{bmatrix}, \; f_i(x_i, u_i, w) := \begin{bmatrix} \alpha_i(u_i + w/n) - \beta_i y \\ 0 \\ 0 \end{bmatrix} \tag{17c}$$

[1] The selection of G_i in (17b) is carried out in such a way to guarantee that map G_i be outer semicontinuous so that the properties established later in Proposition 2 can be concluded.

$$g_i^{\text{on}}(x_i) := \begin{bmatrix} q_i \\ 1 \\ 1 \end{bmatrix} \quad g_i^{\text{off}}(x_i) := \begin{bmatrix} q_i \\ 1 \\ 0 \end{bmatrix} \quad g_i^{0}(x_i) := \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{17d}$$

and $\tilde{\mathcal{D}}_i^0 = \mathcal{D}_i^0 \times \mathbb{R} \times \{-1, 0, 1\}$, where $\mathcal{D}_i^0$ is defined in (3b). Moreover, we select:

$$\begin{aligned} \mathcal{D}_i^{\text{on}} &= \{(x_i, e) : \alpha_i = 0,\ \phi_{\text{on}}(q_i, e) \geq 0,\ \alpha^T \tilde{q} + \tilde{n} q^* \geq \varepsilon/2\} \\ \mathcal{D}_i^{\text{off}} &= \{(x_i, e) : \alpha_i = 1,\ \phi_{\text{off}}(q_i, e) \leq 0,\ \alpha^T \tilde{q} + \tilde{n} q^* \leq -\varepsilon/2\} \\ \mathcal{C}_i &= \overline{\mathcal{D}_i^c} \cap \{(x_i, e) : q_i \geq 0\}, \end{aligned} \tag{18}$$

where $\overline{\mathcal{D}_i^c}$ is the closure of the complement of $\mathcal{D}_i := \mathcal{D}_i^{\text{on}} \cup \mathcal{D}_i^{\text{off}} \cup \tilde{\mathcal{D}}_i^0$. The variable e represents the shared information, which has been introduced in (12a) and will be used to define the control law. The functions associated to the jump sets in (18) are selected as:

$$\phi_{\text{on}}(q_i, e) := \alpha^T \tilde{q} - \tilde{n}\tilde{q}_i - (\tilde{n} + 1)(\varepsilon + \tilde{n} q^*) \tag{19}$$

$$\phi_{\text{off}}(q_i, e) := \alpha^T \tilde{q} - \tilde{n}\tilde{q}_i - (\tilde{n} - 1)(\varepsilon - \tilde{n} q^*), \tag{20}$$

where we recall that $\tilde{q}_i = q_i - q^*$, $\tilde{q} = q - \mathbf{1}q^*$ and $\tilde{n} = n - n^*$, as defined in equation (12a). While a comprehensive understanding of the relevance behind selection (18) can only be gained by appreciating the proof steps in Section 4, an intuition about the meaning of selection (18) follows by noticing that if a state belongs to $\mathcal{D}_i^{\text{on}}$, node i can be switched on, and vice-versa for $\mathcal{D}_i^{\text{off}}$. Then one may notice that (18) involves two algebraic conditions, the first one ensuring that function $\alpha^T \tilde{q}$ is suitably steered toward zero in light of its quantized dynamics (as established in (15)), and the second one ensuring that switching on and off nodes does not induce Zeno behavior. Additional intuitive explanations about the structure of our proof can be found below, after the statement of Theorem 1.

Remark 3. Note that the state β_i multiplies the whole right hand side in the first equation of (3a) but it only multiplies the quantity y in the definition of f_i in the first equation of (17). This simplification is possible because f_i is only evaluated along flows, and no solutions to (3) can flow if $\beta_i = 0$ and $\alpha_i = 1$ (see the definition of $\mathcal{C}_i^\circ$ in (3b)). ⌟

We may now express the dynamics of the overall system as a hybrid dynamical system described by

$$\begin{aligned} \dot{x} &= f(x), \qquad && x \in \mathcal{C} \\ x^+ &\in G(x), \qquad && x \in \mathcal{D}, \end{aligned} \tag{21}$$

where $x = [x_1 \quad x_2 \quad x_3]^T$, and

$$
\begin{aligned}
f(x) &= \begin{bmatrix} f_1(x_1, \kappa_1(x), w) \\ f_2(x_2, \kappa_2(x), w) \\ f_3(x_3, \kappa_3(x), w) \end{bmatrix}, & \mathcal{C} &= \bigcap_{i=1,2,3} \mathcal{C}_i \\
G(x) &= \bigcup_{k=1,2,3:\ x_k \in \mathcal{D}_i} \bar{G}_k(x), & \mathcal{D} &= \bigcup_{i=1,2,3} \mathcal{D}_i,
\end{aligned}
\tag{22}
$$

where $\kappa_i(x)$, $i = 1,\ 2,\ 3$ denotes the right hand side of (13) and

$$
\bar{G}_1(x) = \begin{bmatrix} G_1(x_1, e) \\ x_2 \\ x_3 \end{bmatrix}, \ \bar{G}_2(x) = \begin{bmatrix} x_1 \\ G_2(x_2, e) \\ x_3 \end{bmatrix}, \ \bar{G}_3(x) = \begin{bmatrix} x_1 \\ x_2 \\ G_3(x_3, e) \end{bmatrix}.
$$

The following proposition establishes useful properties of the closed-loop dynamics (21), obtained after implementing the proposed control scheme. These properties are stated in terms of existence of solutions and nature of the solutions in the presence of possible perturbations affecting the dynamics. In particular, the well-posedness result established next ensures, by virtue of [10, §6.3–6.4], that solutions to (21) coincide with possible generalized solutions arising from vanishing perturbations affecting the dynamics. In addition, well posedness implies important fundamental properties such as existence of solutions[2], continuity of solutions with respect to initial conditions and compactness of the solution set (namely, graphically converging solution sequences converge to a solution). See [10, Chapters 5-6] for details.

Proposition 2. *Hybrid dynamics (21) satisfies the hybrid basic conditions in [10, As. 6.5]. Therefore it is well-posed.*

Proof. To prove the hybrid basic conditions we first observe that both sets $\mathcal{C}$ and $\mathcal{D}$ are closed sets, because they are unions and intersections of closed sets. Moreover, the flow map f, which comprises stacking functions f_i in (17c) with u selected as in (13) is single valued and continuous, thereby trivially satisfying the convexity and outer semicontinuity assumptions. Outer semicontinuity of the set-valued map G also follows from the fact that its graph is closed (and then outer semicontinuity follows from [10, Lemma 5.1]). Closedness of the graph of G can be deduced from noticing that the graphs of functions G_i defined in (17c) are the unions of the (closed) graphs of g_i^{on}, g_i^{off} and g_i^0, and then that, according to the definition in (22), the graph of G is the union of the graphs of $\bar{G}_i$, $i = 1, 2, 3$. Local boundedness of f and G follows trivially from local boundedness of f_i and of all the functions composed to form the set-valued map G. Finally, the

[2] Note that existence of solutions follows from well-posedness, together with the fact that the flow set is the closed complement of the jump set (see the last equation in (18)) so that suitable viability conditions hold for the solutions.

second conclusion of the proposition directly follows from [10, Thm 6.30]. □

The following theorem is the main result of this paper.

Theorem 1. *Consider the overall network cluster whose dynamics is governed by equations (13), (17)–(20), equivalently (21). Under Assumption 1, this network cluster meets the three design goals specified in Section 2.4.*

The proof of Theorem 1 is given in Section 4 and requires suitable characterizations of the overall dynamics and using the peculiar selections of jump and flow sets in (18)–(20), in addition to the instrumental results of Proposition 1, Lemma 1 and Property 1.

From an intuitive viewpoint, the scheme is characterized by a peculiar behavior where the activation and deactivation of the nodes is autonomously decided in order to guarantee convergence to the set $[-\varepsilon/2, \varepsilon/2]$ of the quantity $\alpha^T \tilde{q}$ in order to avoid a zeno effect. Indeed, as a byproduct of Lemma 1, this quantity evolves according to the following "quantized" behavior:

$$\alpha^T \dot{\tilde{q}} \;=\; -\tilde{n} y,$$

and enabling or disabling nodes (namely, changing $\tilde{n}$) allows us to shrink it to zero. Note that a larger selection of ε allows reducing the number of switches, however $\alpha^T \tilde{q}$ will be allowed to grow larger during flow, i.e., the active queue lengths q_i will be expected to eventually reach a larger set around q^*. A second part of the scheme, which is perhaps more classical, is the consensus law, operating when solutions flow. This one can be written in terms of the shifted variables $\tilde{q}_i$ and $\tilde{n}$. From equation (14) and since q^* is constant, the dynamics of the shifted variable $\tilde{q}$ can be written as

$$\dot{\tilde{q}} = \; -k_p L(\alpha)\tilde{q} + \alpha \frac{w}{n} - \beta y, \tag{23}$$

which performs the desired load balancing among the queues of the active nodes. The complete proof is given in Section 4.

4. Proof of Theorem 1

To the aim of proving Theorem 1, we will use a few preliminary results (Lemmas 2– 4 below) establishing useful properties of the hybrid solutions. Throughout the proof, we will use the shortcut notation "solutions" to denote *maximal* solutions, namely solutions maximally extended forward in time (see [10, Def. 2.7]). For our system, maximal solutions actually evolve forever (they are "complete") due to the definition of the flow set in (18), which is the closure of the complement of the jump set (relative to the set where the queues

are non-negative). This definition, ensures suitable viability conditions and then that maximal solutions are complete. Note that we cannot perform a Lyapunov proof of the convergence property established in Theorem 1 because the proven convergence result is non-uniform and it may take an arbitrarily long time to reach the desired attractor. Due to this fact, the attractor is unstable (even though globally attractive), thereby highlighting a somewhat interesting peculiarity of the problem under consideration.

Using non-negativity of the queues, the quantities $\tilde{q}_k$ defined in (12b), clearly satisfy

$$\tilde{q}_k \geq -q^*, \ -\tilde{q}_k \leq q^*, \ \forall k \in \mathcal{N}. \tag{24}$$

These inequalities will be useful in several steps of the proofs below.

Lemma 2. *Consider function $\sigma(e) = \alpha^T \tilde{q} + \tilde{n} q^*$. The following holds:*

1. *If a node is switched on, then $\sigma(e) \geq \dfrac{\varepsilon}{2}, \quad \sigma(e^+) \geq \sigma(e)$.*
2. *If a node is switched off, then $\sigma(e) \leq -\dfrac{\varepsilon}{2}, \quad \sigma(e^+) \leq \sigma(e)$.*
3. *If a solution flows with $\tilde{n} = 0$, then $\dot{\sigma}(e) = 0$.*

Proof. *Proof of item 1.* Consider first the case when a node switches on and notice that necessarily $\tilde{n} \in \{0, -1\}$. Indeed, if $\tilde{n} = 1$, all the nodes are already active. Using $\alpha^T \tilde{q} + \tilde{n} q^* \geq \frac{\varepsilon}{2}$ in (18), we get that a node $k \in \mathcal{N}$ switches on only if

$$\sigma(e) \geq \frac{\varepsilon}{2}. \tag{25}$$

Moreover, using the fact that $\alpha_k = 0$, $\alpha_k^+ = 1$, and the queues are non-negative, we get

$$\sigma(e^+) = \alpha^T \tilde{q} + \tilde{q}_k + (\tilde{n} + 1) q^* = \sigma(e) + q_k \geq \sigma(e), \tag{26}$$

namely the function σ does not decrease across switch-on jumps.

Proof of item 2. Consider now the case when a node switches off and notice that necessarily $\tilde{n} \in \{0, 1\}$. Indeed, if $\tilde{n} = -1$, we have for the only node $k \in \mathcal{N}$ satisfying $\alpha_k = 1$:

$$\phi_{\text{off}}(q_k, e) \quad = \quad \alpha^T \tilde{q} + \tilde{q}_k + 2(\varepsilon + q^*) = 2\tilde{q}_k + 2q^* + 2\varepsilon \ = \ 2q_k + 2\varepsilon > 0, \tag{27}$$

where we used the property that the queues are non-negative. Then inequality (27) with (18) implies that no node will switch off unless $\tilde{n} \geq 0$. Using $\alpha^T \tilde{q} + \tilde{n} q^* \leq -\frac{\varepsilon}{2}$ in (18), we get that a node $k \in \mathcal{N}$ switches off only if

$$\sigma(e) \leq -\frac{\varepsilon}{2}. \tag{28}$$

Moreover, using the fact that $\alpha_k = 1$, $\alpha_k^+ = 0$ and non-negativity of the queues, we get

$$\sigma(e^+) = \alpha^T \tilde{q} - \tilde{q}_k + (\tilde{n} - 1) q^* = \sigma(e) - q_k \leq \sigma(e), \tag{29}$$

namely, the function σ does not increase across switch-off jumps.

Proof of item 3. The proof is direct from Lemma 1 (equation (15)) and $\tilde{n} = 0$. □

Lemma 3. *Consider any solution to the hybrid closed loop (21). Then the following holds:*

1. *The solution flows only if it belongs to one of the following sets:*

$$\mathcal{F}_1 := \{n = 1\} \cap \{\alpha^T \tilde{q} - q^* \leq \frac{\varepsilon}{2}\} \cap \{\alpha_k \beta_k = \alpha_k, k = 1, 2, 3\} \tag{30}$$

$$\mathcal{F}_2 := \{n = 2\} \cap \{|\alpha^T \tilde{q}| \leq \varepsilon\} \cap \{\alpha_k \beta_k = \alpha_k, k = 1, 2, 3\} \tag{31}$$

$$\mathcal{F}_3 := \{n = 3\} \cap \{\alpha^T \tilde{q} + q^* \geq -\frac{\varepsilon}{2}\} \cap \{\alpha_k \beta_k = \alpha_k, k = 1, 2, 3\}. \tag{32}$$

2. *If the solution does not belong to $\mathcal{F}_1 \cup \mathcal{F}_2 \cup \mathcal{F}_3$, then it performs at most two jumps and starts flowing afterwards (inside $\mathcal{F}_1 \cup \mathcal{F}_2 \cup \mathcal{F}_3$).*
3. *The set*

$$\mathcal{A}^\circ := \{n = 2\} \cap \{|\alpha^T \tilde{q}| < \varepsilon\} \cap \{\alpha_k \beta_k = \alpha_k, k = 1, 2, 3\}, \tag{33}$$

is strongly [3] *forward invariant and if a solution starts in $\mathcal{A}^\circ$, then it is eventually continuous.*

Proof. Proof of item 1. Note that $\mathcal{F}_i$, $i = 1, 2, 3$ are disjoint because they correspond to different values of n. Then we can analyze three separate cases. Moreover, the necessity of the last intersection in (30)–(32) follows from the definition of D_i^{act} in (3b) that forces β_k to jump to 1 whenever $\alpha_k = 1$ and $\beta_k = 0$, thus preventing flow of the solution. Therefore in the proof we will always consider $\alpha_k \beta_k = \alpha_k$, for all $k = 1, 2, 3$.
(Case $n = 1$) Consider first the case $n = 1$, so that $\tilde{n} = -1$. Assume that the solution does not belong to $\mathcal{F}_1$, namely $\alpha^T \tilde{q} - q^* > \frac{\varepsilon}{2}$. Take any $k \in \mathcal{N}$ such that $\alpha_k = 0$, then using (24) we have $\phi_{\text{on}}(\tilde{q}_k, e) = \alpha^T \tilde{q} + \tilde{q}_k > \frac{\varepsilon}{2}$. Then, using (18) this implies that the solution belongs to the interior of $\mathcal{D}_k^{\text{on}}$ and it must necessarily jump and cannot flow.
(Case $n = 2$) Consider now the case $n = 2$, so that $\tilde{n} = 0$. Assume that the solution does not belong to $\mathcal{F}_2$, namely assume that $\alpha^T \tilde{q} > \varepsilon$ or $\alpha^T \tilde{q} < -\varepsilon$. Two cases may happen: 1) if $\alpha^T \tilde{q} < -\varepsilon$, then we have

$$\phi_{\text{off}}(q_k, e) = \alpha^T \tilde{q} + \varepsilon < 0,$$

where k denotes any of the two active nodes, and from (18) the state belongs to the interior of $\mathcal{D}_k^{\text{off}}$ and the solution must jump (it cannot flow). 2) if $\alpha^T \tilde{q} > \varepsilon$, then we have

$$\phi_{\text{on}}(q_h, e) = \alpha^T \tilde{q} - \varepsilon > 0, \tag{34}$$

[3] According to [10, Def. 6.25], a set is strongly forward invariant if all solutions starting in this set remain in it for all positive times.

where h denotes the only inactive node, from (18) the solution belongs to the interior of $\mathcal{D}_h^{\text{on}}$ and must jump (it cannot flow).

(Case $n = 3$) Finally, consider the case $n = 3$, so that $\tilde{n} = 1$. Assume that the solution does not belong to $\mathcal{F}_3$, namely $\alpha^T \tilde{q} + q^* < -\frac{\varepsilon}{2}$. Then for all $k \in \mathcal{N}$, we may use (24) to get $\phi_{\text{off}}(\tilde{q}_k, e) = \alpha^T \tilde{q} - \tilde{q}_k \leq \alpha^T \tilde{q} + q^* < -\frac{\varepsilon}{2}$. Moreover, with $n = 3$ all nodes are active so $\alpha_k = 1 \ \forall k \in \mathcal{N}$. Therefore, using (18) this implies that the solution belongs to the interior of $\mathcal{D}_k^{\text{off}}$ and must necessarily jump.

Proof of item 2. Using Lemma 2, assume that $\tilde{n} = 0$. Then from (28) (respectively (25)), a node switches off (respectively on) only if $\sigma(e) \leq -\frac{\varepsilon}{2}$ (respectively $\sigma(e) \geq \frac{\varepsilon}{2}$) and $\sigma(e)$ does not increase (respectively, does not decrease) across the corresponding jump. This means that the solution can only jump once from $\tilde{n} = 0$. If $\tilde{n} \neq 0$, then after the first jump it will reach $\tilde{n} = 0$ and subsequently the analysis above applies, so that the solution cannot jump more than twice and then must start flowing. Namely, from item 1, it must belong to $\mathcal{F}_1 \cup \mathcal{F}_2 \cup \mathcal{F}_3$.

Proof of item 3. Using $\tilde{n} = 0$, denoting by i the only inactive node and by h any of the two active nodes, we have:

$$\begin{aligned} \phi_{\text{on}}(q_i, e) &= \alpha^T \tilde{q} - \varepsilon < 0, \\ \phi_{\text{off}}(q_h, e) &= \alpha^T \tilde{q} + \varepsilon > 0, \end{aligned}$$

and from (18), the solution does not belong to either $\mathcal{D}_i^{\text{on}}$ or $\mathcal{D}_h^{\text{off}}$. Regarding $\tilde{\mathcal{D}}_k^0$, $k = 1, 2, 3$, since $\alpha_k \beta_k = \alpha_k$, for all $k = 1, 2, 3$, then a single jump may occur from D_i^{empty} for the (only) inactive node i. This possible jump does not cause any variation of $\alpha^T \tilde{q}$ (because α_i remains zero). Strong forward invariance then comes from Lemma 1 (equation (15)) stating that along flows $\alpha^T \tilde{q}$ remains constant in this set. As a consequence, the solution is eventually continuous. □

Lemma 4. *The following holds:*

1. *If a solution flows in $\mathcal{F}_1$, then after a finite time it jumps to*

$$\mathcal{A}_{>0} := \mathcal{F}_2 \cap \left\{ \sigma(e) > \frac{\varepsilon}{2} \right\} = \mathcal{F}_2 \cap \left\{ \alpha^T \tilde{q} > \frac{\varepsilon}{2} \right\},$$

   and either flows forever in $\mathcal{A}_{>0}$ or it jumps, after a finite time, to $\mathcal{F}_3$, where it starts flowing again.

2. *If a solution flows in $\mathcal{F}_3$, then after a finite time it jumps to*

$$\mathcal{A}_{<0} := \mathcal{F}_2 \cap \left\{ \sigma(e) < -\frac{\varepsilon}{2} \right\} = \mathcal{F}_2 \cap \left\{ \alpha^T \tilde{q} < -\frac{\varepsilon}{2} \right\},$$

   and either flows forever in $\mathcal{A}_{<0}$ or it jumps, after a finite time, to the set

$$\tilde{\mathcal{F}}_1 = \mathcal{F}_1 \cap \left\{ \mathbf{1}^T \tilde{q} \leq -q^* - \frac{\varepsilon}{2} \right\}, \tag{35}$$

   where it starts flowing again.

3. *If a solution flows from $\tilde{\mathcal{F}}_1$, then after a finite time, it jumps to the set $\mathcal{A}^\circ$ and it is eventually continuous.*

Proof. *Proof of item 1.* Since the solution flows in $\mathcal{F}_1$, no node can deactivate, indeed only one node is active and following the reasoning at the beginning of the proof of item 1 of Lemma 2, it cannot deactivate. Moreover, (from (30)), we have $\alpha^T\tilde{q} - q^* \leq \frac{\varepsilon}{2}$, and since $\tilde{n} = -1$ and using Lemma 1 (equation (15)) we get

$$\frac{d}{dt}(\alpha^T\tilde{q}) = -\tilde{n}y = y,$$

which implies that $\alpha^T\tilde{q}$ is strictly increasing. Finally, the solution will not jump until $\alpha^T\tilde{q} = -\tilde{n}q^* + \frac{\varepsilon}{2} = q^* + \frac{\varepsilon}{2}$ because otherwise (from (18)) it would not belong to $\mathcal{D}_i^{\text{on}}$ for any $i \in \mathcal{N}$. Note that, from (27) the active node can not deactivate. On the other hand, when

$$\alpha^T\tilde{q} = q^* + \frac{\varepsilon}{2}, \tag{36}$$

one node will certainly activate because otherwise the solution would flow outside $\mathcal{F}_1$ and this contradicts Lemma 3 *item 1.* Denote by $k \in \mathcal{N}$ the node that activates. Then $\phi_{\text{on}}(q_k, e) \geq 0$ and from (24) the solution jumps to

$$(\alpha^T\tilde{q})^+ = \alpha^T\tilde{q} + \tilde{q}_k = q^* + \frac{\varepsilon}{2} + \tilde{q}_k \geq \frac{\varepsilon}{2}, \tag{37}$$

$$(\tilde{n})^+ = 0. \tag{38}$$

Now, one of the following applies:

1. The solution jumps to $\mathcal{A}_{>0}$ where it flows forever.
2. The solution jumps to $\mathcal{A}_{>0}$, it flows for some time and then jumps again, in which case it must jump to $\mathcal{F}_3$ and start flowing. Indeed from Lemma 2, after the first jump we have $\sigma(e^+) \geq \sigma(e) \geq \frac{\varepsilon}{2}$ and $\sigma(e)$ remains constant when flowing in $\mathcal{A}_{>0}$ (where $\tilde{n} = 0$), therefore from Lemma 2 the subsequent jump may only be an activation and the solution must then jump to $\mathcal{F}_3$ and start flowing (from Lemma 3, *item 1*).
3. The solution jumps outside $\mathcal{A}_{>0}$, namely, also using (37) (which implies that $(\alpha^T\tilde{q})^+$ is positive), we have $(\alpha^T\tilde{q})^+ > \varepsilon$. This implies that the third node i activates, because the solution belongs to the interior of $\mathcal{D}_i^{\text{on}}$ (see (18)). Indeed, using $\tilde{n}^+ = 0$, we obtain:

$$(\alpha^T\tilde{q} + \tilde{n}q^*)^+ = (\alpha^T\tilde{q})^+ > \varepsilon > \frac{\varepsilon}{2}$$

and

$$\phi_{\text{on}}(q_i, e^+) = (\alpha^T\tilde{q})^+ - \varepsilon > 0.$$

After this second activation, we have $(\tilde{n})^{++} = 1$ and $(\alpha^T \tilde{q})^{++} = (\alpha^T \tilde{q})^+ + \tilde{q}_i > \varepsilon - q^*$ (where we also used (24) and we use superscript "++" to denote the value after the second jump) and then the solution belongs to $\mathcal{F}_3$ (see equation (32) in Lemma 3, *item 1*).

Proof of item 2. Since the solution flows in $\mathcal{F}_3$, all nodes are active. Moreover from (32) we have $\alpha^T \tilde{q} + q^* \geq -\frac{\varepsilon}{2}$; recalling that $\tilde{n} = 1$ and using Lemma 1 (equation (15)), we get

$$\frac{d}{dt}(\alpha^T \tilde{q}) = -\tilde{n}y = -y,$$

which implies that $\alpha^T \tilde{q}$ is strictly decreasing. Moreover, the solution will not jump until $\alpha^T \tilde{q} = -\tilde{n}q^* - \frac{\varepsilon}{2} = -q^* - \frac{\varepsilon}{2}$ because otherwise (from (18)) it would not belong to $\mathcal{D}_i^{\text{off}}$ for any $i \in \mathcal{N}$. Note that no node can activate because all nodes are active. On the other hand, when

$$\mathbf{1}^T \tilde{q} = \alpha^T \tilde{q} = -q^* - \frac{\varepsilon}{2}, \tag{39}$$

one node will certainly deactivate because otherwise the solution would flow outside $\mathcal{F}_3$ and this contradicts again Lemma 3, *item 1*. Denote by $k \in \mathcal{N}$ the node that deactivates. Then $\phi_{\text{off}}(q_k, e) \leq 0$ and from (24) the solution jumps to

$$(\alpha^T \tilde{q})^+ = \alpha^T \tilde{q} - \tilde{q}_k = -q^* - \frac{\varepsilon}{2} - \tilde{q}_k \leq -\frac{\varepsilon}{2}, \tag{40}$$

$$(\tilde{n})^+ = 0. \tag{41}$$

Now, one of the following applies:

1. The solution jumps to $\mathcal{A}_{<0}$ where it flows forever.
2. The solution jumps to $\mathcal{A}_{<0}$, it flows for some time and then jumps again, in which case it must jump to $\mathcal{F}_1$ and start flowing. Indeed from Lemma 2, after the first jump we have $\sigma(e^+) \leq \sigma(e) \leq -\frac{\varepsilon}{2}$ and $\sigma(e)$ remains constant when flowing in $\mathcal{A}_{<0}$ (where $\tilde{n} = 0$), therefore from Lemma 2 the subsequent jump may only be an activation and the solution must then jump to $\mathcal{F}_3$ and start flowing (from Lemma 3, *item 1*). In addition, the solution must start flowing from $\tilde{\mathcal{F}}_1$, indeed before the first jump, it satisfied (39), $\mathbf{1}^T \tilde{q}$ remains constant across jumps, and when flowing in $\mathcal{A}_{<0}$ (denoting by h the only inactive node) we have (also from Lemma 1, equation (15) and (17c) with $\alpha_h = 0$),

$$\mathbf{1}^T \dot{\tilde{q}} = \alpha^T \dot{\tilde{q}} + \dot{\tilde{q}}_h \leq 0.$$

3. The solution jumps outside $\mathcal{A}_{<0}$, namely, also using (40), $(\alpha^T \tilde{q})^+ < -\varepsilon$. This implies that a second node i deactivates because the solution belongs to the interior of $\mathcal{D}_i^{\text{off}}$ (see (18)). Indeed, from Lemma 2, after the first jump we have

$$(\alpha^T \tilde{q} + \tilde{n}q^*)^+ = (\alpha^T \tilde{q})^+ < -\varepsilon < -\frac{\varepsilon}{2}$$

and

$$\phi_{\text{off}}(q_i, e^+) = (\alpha^T \tilde{q})^+ + \varepsilon < 0.$$

After this second deactivation, we have $(\tilde{n})^{++} = -1$ and $(\alpha^T \tilde{q})^{++} = (\alpha^T \tilde{q})^+ - \tilde{q}_i < -\varepsilon + q^*$ (where we also used (24)) and then the solution belongs to the set (35) where the right most constraint on $\mathbf{1}^T \tilde{q}$ comes from the fact that the queue lengths does not change across jumps and that before the first jump we had (39), which implies $\tilde{q}_i + \tilde{q}_h + \tilde{q}_k = \mathbf{1}^T \tilde{q} = -q^* - \frac{\varepsilon}{2}$. Since the solution belongs to $\tilde{\mathcal{F}}_1$ (from Lemma 4, *item 3*).

Proof of item 3. Since $\tilde{\mathcal{F}}_1 \subset F_1$, then *item 1* implies that the solution jumps to $\mathcal{A}_{>0}$ after a finite time. We show below that it actually jumps to $\mathcal{A}_{>0} \cap \mathcal{A}^\circ$ (see (33)) and then it is eventually continuous from Lemma 3, *item 3*. To this aim, denote by k the active node and by i, h the inactive nodes when flowing in $\mathcal{F}_1$. Denote by $(t_j, j-1)$ the hybrid time when the solution jumps to $\mathcal{A}_{>0}$ by activating node i. Using equation (36) in the proof of *item 1*, we have

$$\alpha^T(t_j, j-1)\tilde{q}(t_j, j-1) = \tilde{q}_k(t_j, j-1) = q^* + \frac{\varepsilon}{2}. \tag{42}$$

Moreover, we show below that $q_i(t_j, j-1) = q_h(t_j, j-1) = 0$, which implies

$$\alpha^T(t_j, j)\tilde{q}(t_j, j) = \tilde{q}_k(t_j, j) = q^* + \frac{\varepsilon}{2} - q^* = \frac{\varepsilon}{2} < \varepsilon.$$

which proves that the solution belongs to $\mathcal{A}^\circ$. It remains to show that $q_i(t_j, j-1) = q_h(t_j, j-1) = 0$ (both queues are empty at time $(t_j, j-1)$). We show this by contradiction assuming that at least one queue (that we may denote q_i) is non-empty. In particular, if $q_i(t_j, j-1) \neq 0$, then from (17c), we had $\beta_i = 1$ all along the flowing in $\mathcal{F}_1$, which implies (keeping in mind that k is the only active node and using Lemma 1, equation (15)) and (17c)),

$$\mathbf{1}^T \dot{\tilde{q}} = \dot{\tilde{q}}_k + \dot{\tilde{q}}_i + \dot{\tilde{q}}_h = \underbrace{\alpha^T \dot{\tilde{q}}}_{y} + \underbrace{\dot{\tilde{q}}_i}_{-y} + \underbrace{\dot{\tilde{q}}_h}_{0} \leq 0. \tag{43}$$

Since the solution started flowing in $\tilde{\mathcal{F}}_1$, where $\mathbf{1}^T \tilde{q} \leq -q^* - \frac{\varepsilon}{2}$ (see (35)) and $\mathbf{1}^T \tilde{q}$ remains constant across jumps, then using the non-increase property implied by equation (43), we obtain that at the jump time $(t_j, j-1)$ we have

$$\tilde{q}_k(t_j, j-1) + \tilde{q}_i(t_j, j-1) + \tilde{q}_h(t_j, j-1) = \mathbf{1}^T \tilde{q}(t_j, j-1) \leq -q^* - \frac{\varepsilon}{2},$$

which can be combined with equation (42) to obtain

$$\tilde{q}_i(t_j, j-1) + \tilde{q}_h(t_j, j-1) \leq -2q^* - \varepsilon.$$

This last relationship contradicts (24) (namely, all queue lengths can never be negative), thus establishing a contradiction and completing the proof. □

Lemma 5. *All solutions to* (21) *eventually evolve continuously in set* $\mathcal{A} = \mathcal{F}_2 \cap \left\{|\alpha^T \tilde{q}| \leq \varepsilon\right\}$.

Proof. We prove the result using Lemma 2, 3 and 4. Consider any solution to (21) and the following applies

1. from Lemma 3 *item 2* we know that it converges in finite time to $\mathcal{F}_1 \cup \mathcal{F}_2 \cup \mathcal{F}_3$ where it starts flowing.
2. If it belongs to $\mathcal{A}$, it may flow forever (in which case the proof is completed) or it may jump and two cases occur:
   (a) one node activates and from Lemma 2 we have $\sigma(e^+) \geq \sigma(e) \geq \frac{\varepsilon}{2}$ so that from the same lemma no node can deactivate. Then the solution must flow and from Lemma 3 (*item 1*) it must belong to $\mathcal{F}_3$ and Lemma 4 applies.
   (b) one node deactivates and the same reasoning can be followed to conclude that the solution must flow in $\mathcal{F}_1$ and Lemma 4 applies.
3. From Lemma 4 the solution evolves according to the following scheme:

$$\mathcal{F}_1 \to \mathcal{A}_{>0} \to \mathcal{F}_3 \to \mathcal{A}_{<0} \to \tilde{\mathcal{F}}_1 \to \mathcal{A}^\circ. \tag{44}$$

   and cannot flow forever neither in $\mathcal{F}_1$, nor in $\mathcal{F}_3$ or in $\tilde{\mathcal{F}}_1 \subset \mathcal{F}_1$. Therefore, it necessarily flows forever in $\mathcal{A}_{>0} \subset \mathcal{A}$, in $\mathcal{A}_{<0} \subset \mathcal{A}$, or finally in $\mathcal{A}^\circ \subset \mathcal{A}$, which completes the proof.

□

Proof of Theorem 1. The processing constraint (11) is proven in Proposition 1.

Next we prove the convergence properties in (8)–(10). To this aim, based on Lemma 5, we know that any solution to (21) is eventually continuous. After the solution evolves continuously, thereby not exhibiting any jumps and satisfying $n = n^* = 2$, we may use equation (23) to obtain that function $V(\alpha, \tilde{q}) = \tilde{q}^T L(\alpha)\tilde{q}$ satisfies along flows:

$$\dot{V} = \tilde{q}^T L(\alpha)(-k_p L(\alpha)\tilde{q} + \alpha\frac{w}{n} - \beta y) = -k_p \tilde{q}^T L(\alpha)^2 \tilde{q} + \tilde{q}^T \underbrace{L(\alpha)(\alpha\frac{w}{n} - \beta y)}_{=0}, \tag{45}$$

where the term on the right is equal to zero due to Property 1. Then, from the structure of $L(\alpha)$ and standard consensus results [15], we obtain that all active nodes reach consensus. In particular, since from Lemma 5 the solution eventually evolves continuously in set $\mathcal{F}_2$, then we have

$$\lim_{t\to\infty} n(t, j^*) = 2 = n^*,$$

where j^* is the maximum j in the eventually continuous domain of the solution. Moreover, notice that after a finite time, when the solution evolves continuously in $\mathcal{F}_2$, we have that $\alpha^T \tilde{q}$ remains constant (from

Lemma 1, equation (15)) and that $|\alpha^T \tilde{q}| \leq \varepsilon$ (from the definition of $\mathcal{F}_2$). This, together with the consensus established by (45), implies that if α_k converges to 1, then

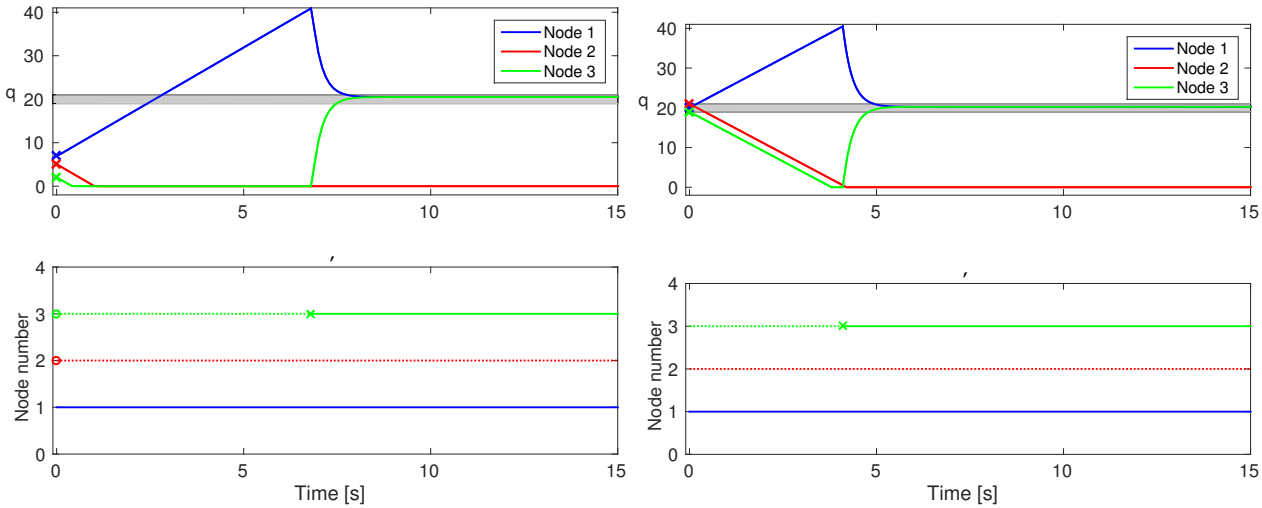
$$\lim_{t\to\infty} |\tilde{q}_k(t, j^*)| = \frac{1}{2} \lim_{t\to\infty} |\alpha^T \tilde{q}| \leq \frac{\varepsilon}{2},$$

and if α_k converges to 0, then the corresponding node remains inactive and its queue q_k converges to zero in finite time. □

5. Simulations

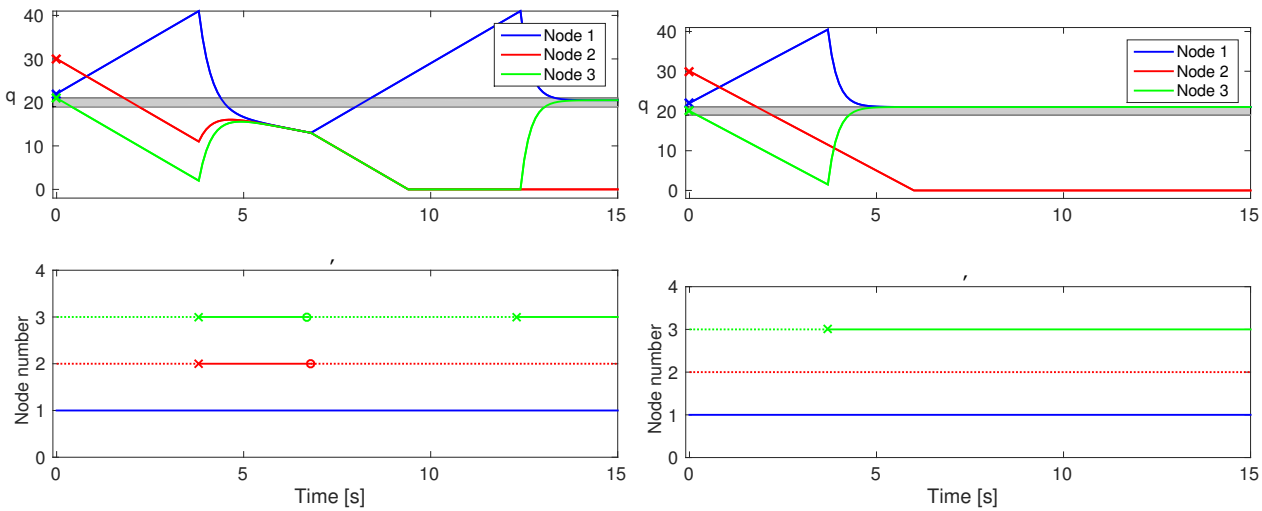
Simulations have been performed in MATLAB/Simulink with the Hybrid Equations (HyEQ) Toolbox [18]. A first set of simulations has been performed on the three-node cluster analyzed in the paper and corresponds to the results in Figure 2. Then, the behavior of the proposed algorithm in a larger cluster of 20 nodes is verified in a more general setting that goes beyond the restrictive assumptions of our main theorem. The resulting responses are shown in Figure 3. Detailed comments follow.

In our first set of simulations we selected a service rate $y = 5$, the reference for the queue length is $q^* = 20$ and the parameter of the load balancing control law in (13) is $k_p = 2$. Figure 2 shows four situations when the request rate is constant and is chosen as $w = 2y = 10$ (according to Assumption 1). Figures 2a,b,c,d show the evolution of the three queues and the components of α, from various initial conditions when selecting $\varepsilon = 2$. In the upper plot of each figure, the gray shaded area denotes the interval $\left[q^* - \frac{\varepsilon}{2}, q^* + \frac{\varepsilon}{2}\right] = [19, 21]$. In the lower plot we see one horizontal line for each node. Whenever the line is solid, the node is active. Conversely a dotted line means that the node is inactive. Upon activation, we report an $\times$ on the corresponding trace and upon deactivation we report an $\circ$. The four simulations of Figure 2 are commented below. In Figure 2a, all nodes are initially active and all the queues are below the desired value q^*. One can see that the solution initially belongs to $\tilde{\mathcal{F}}_1$ and, as established in Lemma 4, it jumps around ordinary time $t = 7$ to the set $\mathcal{A}^\circ$ in (33) where it remains forever. Instead in Figure 2b, there is initially only one active node and all queues are close to q^*. This means that the solution belongs to $\mathcal{F}_3$ and, according to Lemma 5, it flows for some time in $\mathcal{F}_3$ and around ordinary time $t = 4$ it jumps to $\mathcal{F}_2$ from where it converges to $\mathcal{A}^\circ$ where it remains forever. Figures 2c and 2d show two situations with similar initial conditions. In both cases, the solution initially lies in $\mathcal{F}_1$. However, one can see that the evolutions are radically different: in Figure 2c the solution starts in $\mathcal{F}_1$ and then flows and jumps through the sets in (44). Indeed, it jumps to $\mathcal{A}_{>0}$ and then immediately to $\mathcal{F}_3$ (both nodes 2 and 3 instantaneously switch on) around time $t = 4$. Then around time $t = 7$ both these nodes go off again and the solution jumps to $\mathcal{A}_{<0}$ and $\tilde{\mathcal{F}}_1$. Finally, around ordinary time $t = 12$, the solution jumps to $\mathcal{A}^\circ$ where it remains forever. Conversely, in Figure 2d, the solution evolves according to



(a) $q(0,0) = (7,5,2)^T$, $\alpha(0,0) = (1,1,1)^T$.

(b) $q(0,0) = (20,21,19)^T$, $\alpha(0,0) = (1,0,0)^T$.



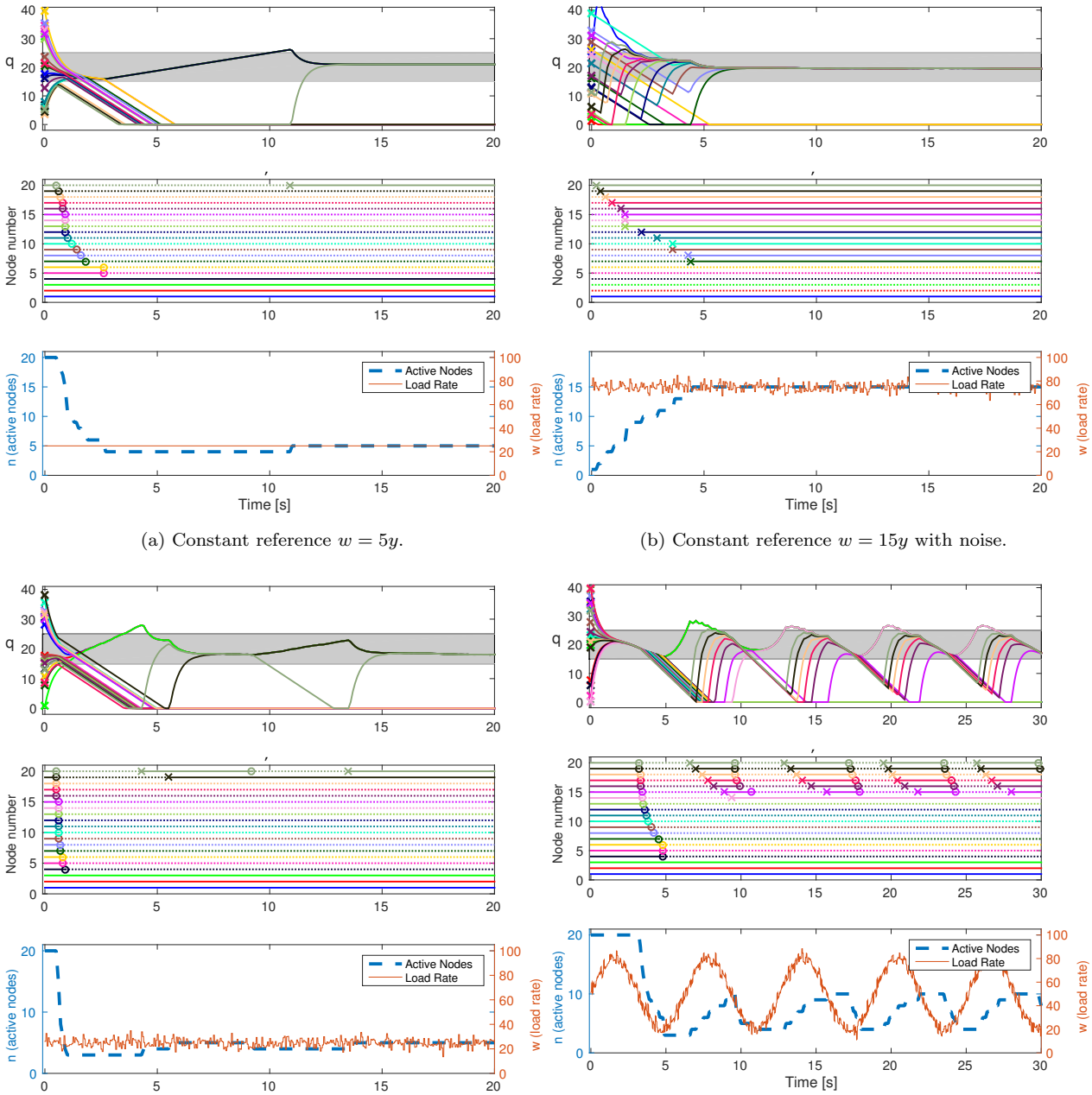
(c) $q(0,0) = (22,30,21)^T$, $\alpha(0,0) = (1,0,0)^T$.

(d) $q(0,0) = (22,30,20)^T$, $\alpha(0,0) = (1,0,0)^T$.

Figure 2: Simulations of the three-node cluster representing the evolution of the queues q_i, $i = 1,2,3$ with $q^* = 20$, $n^* = 2$, $k_p = 2$, $w = 10$, $y = 5$ and $\varepsilon = 2$, from different initial conditions.

the scenario in item 1 of Lemma 4. Namely around ordinary time $t = 4$ it jumps from $\mathcal{F}_1$ to $\mathcal{A}_{>0}$ where it flows forever, because $\frac{\varepsilon}{2} < \sigma(e) < \varepsilon$. One may appreciate from the lower traces of Figure 2 that switching on/off servers is performed in a fair way, avoiding excessive switching events, which is an advantage of the proposed algorithm.

In Figure 3 we illustrate the behavior of the proposed controller in a much more general setting that goes beyond the cases where for which Theorem 1 guarantees the goals specified in Section 2.4. In particular, we



(a) Constant reference $w = 5y$.

(b) Constant reference $w = 15y$ with noise.

(c) Constant reference $w = 5y$ with noise.

(d) Sinusoidal reference with noise.

Figure 3: Simulations of a 20-node cluster with several time varying requests $w(t)$, $y = 5$ and $\varepsilon = 10$, from four different random initial conditions.

note that nowhere in the specification of our control strategy we make explicit reference to the fact that only three nodes are addressed, therefore we may apply the control solution to any number of nodes. We focus

here on a cluster of $N = 20$ nodes and assume that this cluster is associated to a fully connected undirected graph. Moreover, instead of assuming a fixed rate w coming from the source, we address a general situation with a time-varying request $t \mapsto w(t)$ entering the cluster. Due to this, we generalize quantities n^* and $\tilde{n}$ (defined for the three-node case in equations (1) and (12b)) to the following ones:

$$n^* = \text{round}\left(\frac{w}{y}\right), \quad \tilde{n} = \text{sat}(n - n^*), \tag{46}$$

where sat is the unit saturation function. The selection in (46) reduces to the previous definitions whenever $N = 3$ (three-node cluster). Note that since w is not constant, also quantity n^* is non-constant and corresponds to the closest integer to the necessary number of servers to satisfy the instantaneous request rate.

In all the simulations of Figure 3 we keep the same values $k_p = 2$, $y = 5$, $q^* = 20$, as in Figure 2, but due to the presence of noise in the request w, we select the larger value $\varepsilon = 10$ to avoid too frequent switching triggered by noise. Moreover, in all the simulation we choose random initial conditions between 0 and $40 = 2q^*$ for the initial queue lengths. The top and middle plots coincide with those of Figure 2 (with matching colors for the same nodes) while the lower plot indicates the instantaneous number n of active nodes (blue dashed – left scale) together with the request rate w (brown solid – right scale). This last curve, when using the left scale, corresponds to quantity n^* defined in (46) before the rounding. Therefore we expect to see simulations where the queue lengths of the active nodes remain within (as close as possible to) the shaded interval $\left[q^* - \frac{\varepsilon}{2}, q^* + \frac{\varepsilon}{2}\right]$ and the lower dashed curve (n) converges to the brown one (n^*). Figure 3a shows a first result starting from all nodes active and with a constant request $w = 5y$. One can clearly see the generalization of the patterns in Figure 2 ensuring suitable convergence. Similarly, Figure 3c shows the same setting with band limited Gaussian noise added to w. One may see that the noise does not cause persistent switch on/off of the nodes, which is a desirable feature. Figure 3b shows the converse situation where all but one nodes are off and $w(t) = 15y + d(t)$, with d being the same Gaussian noise. Here we may appreciate a gradual activation of the the nodes up to the necessary number of 15 of them, without any overshoot. A final test is ran in Figure 3d where we have a more complex scenario with very large oscillations in w and superimposed noise. One may clearly see a desirable activation/deactivation pattern that manages keeping the queues of nodes 1, 2, 3 and 14 (blue, red, green and pink) in a desirable range, while turning periodically on and off the last six nodes.

6. Conclusions and future work

In this paper, a novel control architecture for a three-nodes network cluster has been proposed. The control law, which is based both on consensus algorithms and on an appropriate hybrid modeling approach,

allows managing the activity of each node in a distributed manner but with the knowledge of two scalar quantities shared within the cluster. This work can be seen as a first set of results already showing great potential behind the adopted hybrid control paradigm in a very simple setting. In future work, the generality of our formulation will be exploited to extend the proposed approach to more general scenarios. These generalizations may comprise the consideration of a robust paradigm with respect to delays or packet losses, or even more, the extension to larger clusters, distributed architectures based on local information only and in the presence of multiple and more complex sources producing non-constant computational requests. Finally, more realistic simulations in the Network Simulator NS2 could be performed.

Acknowledgments

This work is supported in part by the ANR project LimICoS contract number 12 BS03 005 01, by the iCODE institute, research project of the Idex Paris-Saclay, grant OptHySYS funded by the University of Trento, and by the PEPS JCJC project HYBRISCON nb. 52 814.

References

[1] B.B. Alagoz, A. Kaygusuz, and A. Karabiber. A user-mode distributed energy management architecture for smart grid applications. *Energy*, 44(1):167–177, 2012.

[2] E. Altman and H. Kameda. Equilibria for multiclass routing in multi-agent networks. In *Proc. of the 40th IEEE Conference on Decision and Control*, volume 1, pages 604–609, 2001.

[3] M. Berekmery, D. Serrano, S. Bouchenak, N. Marchand, and B. Robu. A control approach for performance of big data systems. In *Proc. of the IFAC wolrd congress*, pages 152–157, 2014.

[4] C. De Persis and P. Frasca. Hybrid coordination of flow networks. In J. Daafouz, S. Tarbouriech, and M. Sigalotti, editors, *Hybrid Systems with Constraints*, pages 121–143. Wiley, 2013.

[5] C. De Persis and P. Frasca. Robust self-triggered coordination with ternary controllers. *IEEE Trans. on Automatic Control*, 58(12):3024–3038, 2013.

[6] C. De Persis and C.S. Kallesoe. Pressure regulation in nonlinear hydraulic networks by positive and quantized controls. *IEEE Trans. on Control Systems Technology*, 19(6):1371–1383, 2011.

[7] M. Franceschelli, A. Giua, A. Pisano, and E. Usai. Finite-time consensus for switching network topologies with disturbances. *Nonlinear Analysis: Hybrid Systems*, 10:83–93, 2013.

[8] K. Gilly, C. Juiz, and R. Puigjaner. An up-to-date survey in web load balancing. *World Wide Web*, 14(2):105–131, 2011.

[9] R. Goebel, R. Sanfelice, and A.R. Teel. Hybrid dynamical systems. *IEEE Control Systems Magazine*, 29(2):28–93, April 2009.

[10] R. Goebel, R.G. Sanfelice, and A.R. Teel. *Hybrid Dynamical Systems: modeling, stability, and robustness*. Princeton University Press, 2012.

[11] F. Gouaisbaut, I. Queinnec, and S. Tarbouriech. Stability of load balancing control. In John Chiasson and Jean-Jacques Loiseau, editors, *Applications of Time Delay Systems*, volume 352 of *Lecture Notes in Control and Information Sciences*, pages 77–95. Springer Berlin Heidelberg, 2007.

[12] H. Kameda, E. Fathy, I. Ryu, and J. Li. A performance comparison of dynamic vs. static load balancing policies in a mainframe-personal computer network model. In *Proc. of the 39th IEEE Conference on Decision and Control*, volume 2, pages 1415–1420, 2000.

[13] H. Kameda, J. Li, C. Kim, and Y. Zhang. *Optimal load balancing in distributed computer systems*. Springer Publishing Company, Incorporated, 2011.

[14] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall. Reducing network energy consumption via sleeping and rate-adaptation. In *Symposium on Networked Systems Design and Implementation*, volume 8, pages 323–336, 2008.

[15] R. Olfati-Saber, J. A. Fax, and R.M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[16] R. Olfati-Saber and R.M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. on Automatic Control*, 49(9):1520–1533, 2004.

[17] W. Ren and R.W. Beard. *Distributed Consensus Algorithm in Multi-Vehicle Cooperative Control*. Princeton University Press, 2012.

[18] R. G. Sanfelice, D. Copp, and P. A Nanez. A toolbox for simulation of hybrid systems in Matlab/Simulink: Hybrid equations (HyEQ) toolbox. In *Hybrid Systems: Computation and Control Conference*, pages 101–106, 2013.

[19] R. Sipahi, S.I. Niculescu, C.T. Abdallah, W. Michiels, and K. Gu. Stability and stabilization of systems with time delay. *IEEE Control Systems Magazine*, 31(1):38–65, 2011.

[20] Z. Tang, J. D. Birdwell, J. Chiasson, C. T. Abdallah, and M. M. Hayat. Closed loop control of a load balancing network with time delays and processor resource constraints. In *Advances in Communication Control Networks*, pages 245–268. ser. Lecture Notes in Control and Information Sciences, S. Tarbouriech, C. Abdallah, and J. Chiasson, Eds. Springer, 2005.

[21] M. H. Willebeek-LeMair and A.P. Reeves. Strategies for dynamic load balancing on highly parallel computers. *IEEE Trans. on Parallel and Distributed Systems,*, 4(9):979–993, 1993.

[22] S.W. Wong, L. Valcarenghi, S.H. Yen, D.R. Campelo, S. Yamashita, and L. Kazovsky. Sleep mode for energy saving pons: advantages and drawbacks. In *Proc. of the IEEE Global Communication conference workshops*, pages 1–6, 2009.