An Optimal General Purpose Scheduler for Networked Control Systems

Ahmed Elmahdi*, Ahmad F Taha*, Dengfeng Sun†, and Jitesh H. Panchal‡

***Abstract*—The network-induced time delay is one of the major challenges that appear when the design or the analysis of any Networked Control System (NCS) is needed. The delay can be tackled in the network level by performing control of network actions. Since the process of prioritizing the network access is the main contributor of the network-induced time delay, then the efficient way to control the time delay is to improve the scheduling criterion. This paper proposes an optimal scheduling criterion that improves the quality of service for each node in the network and at the same time maintains the stability of the control system. The proposed optimal scheduling protocol is a hybrid scheduling criterion that combines the advantages of the dynamic and the static scheduling protocols. First, the algorithm of the optimal scheduling is discussed. Second, the optimal control-scheduling problem is formulated. The mixed logical-dynamical optimization problem is then solved. Finally, numerical examples and simulation results are introduced and discussed.**

***Index Terms*—Networked Control Systems, Schedulers, Hybrid Scheduling, Traffic Division Arbitration, TDA.**

I. Introduction

CONTRARY to all prior research in the field of Networked Control Systems (NCS), which considers either dynamic or static schedulers to prioritize the network access, in this paper we advocate the use of the novel hybrid scheduler, named Traffic-Division Arbitration (TDA) as proposed [8]. It can handle different situations and respond adequately to all requirements. This flexibility comes from the fact that TDA protocol alternates between dynamic and static scheduling throughout the scheduling process by dividing the traffic into transmission cycles of two levels of arbitration. However, in large scale systems like the smart-grid, there is a need to optimize the behavior of the TDA scheduler to make it able to pick up the optimal scheduling criterion during the operation of the system. We are exclusively the first to introduce a general purpose scheduler in the field of NCS that can optimize the network access based on the desired performance during the system operation.

Generally, when we have shared resources between multiple users, the use of a scheduling protocol becomes a necessity. In this paper, we are extending the TDA scheduling algorithm to serve as a general purpose scheduler. As depicted in [8], this scheduler divides the network traffic into cycles based on a predetermined static threshold. Previously, this threshold was computed by trial and error criterion. The objective of this paper is to find a systematic way to compute the threshold in order to make the TDA a general purpose scheduler with optimal threshold. To achieve this goal, we form an dynamic optimal control problem. The solution of this optimal control problem provides a dynamic optimal threshold that achieves a pre-described objective. The TDA scheduling protocol can be used in a wide range of applications. If the application requires the scheduling protocol to be fair and statically assign an access order for each node, then this can be achieved by using a low error threshold. On the other hand, it can be adjusted for some NCS applications where the network access needs to be prioritized. This unfair dynamic behavior can be achieved by setting a high threshold.

The organization of this paper is as follows. In Section II, we provide a literature review about the recent research trends in the area of scheduling protocols for NCSs, in addition to the research problem addressed in this paper. The general purpose scheduler, namely the optimal TDA, is introduced in Section III with a schematic example that exemplifies the protocol's operation. In Section IV, we formulate the optimal control problem that finds the optimal threshold. Section V includes numerical simulations that shows the usefulness of the proposed optimized TDA protocol, in comparison with another well-known protocol. Conclusions and closing remarks are included in Section VI.

II. Literature Review and Research Gaps

Since the emerging of the networked dynamical systems, there is a plethora of research results on scheduling the access to the shared communication channel in control systems. As an example, the first scheduling protocol to be used in practice was the controller area network protocol which mainly was produced to schedule control events and sensing measurements in automobile industry [1] and [2]. It has quickly evolved to become an important element for building any automated manufacturing systems with a wide range of different versions [3]–[5]. The dynamic scheduler Try-Once Discard (TOD) protocol was introduced in [6] and [7]. A clear problem with dynamic schedulers is that we cannot force a fair share of the communication channel [8].

Static schedulers like Round-Robin (RR) protocol and Token-Ring (TR) protocol have a lot of applications

*Graduate Student, School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana 47907 (aelmahdi,tahaa)@purdue.edu

†Assistant Professor, School of Aeronautics and Astronautics, Purdue University, West Lafayette, Indiana 47907 dsun@purdue.edu

‡Assistant Professor, School of Mechanical Engineering, Purdue University, West Lafayette, Indiana 47907 panchal@purdue.edu

because their algorithms are easy to implement and both exhibit deterministic behavior. The main drawback of static scheduling in the control environment is that we cannot guarantee stability with fixed intervals between control system events [6] and [9]. More recent results appeared in [10]–[14] where the authors introduced a class of Lyapunov uniformly globally asymptotically stable protocols for NCS. The authors provided a framework that can be used to generate new scheduling protocols that preserve L_p stability when we choose sufficiently small design parameters.

In [15]–[17], the authors proposed a scheduling method where the bandwidth allocation and the sampling period determined to handle periodic data, sporadic data and message data. They proposed a scheduler that handles different types of data, but it doesn't handle different objectives of the system operation to tackle conflicting objectives like fairness and urgency. The authors in [17] investigated how the NCS performance can be adjusted so that the quality-of-service (QoS) can be guaranteed. The main objective of their approach was to give the required QoS priority over the system performance in general. In spite of being able to achieve the desired operation and keep the intended QoS, the proposed approach can degrade the performance.

Unfortunately, none of the scheduling protocols of the NCS literature is general enough to handle large-scale systems requirements like smart grid systems. Our proposed optimized general purpose scheduler is aimed to be able to generate online static or dynamic schedules as desired. It is intended to optimally handle and meet the desired operation requirements and guarantee the QoS by using the same algorithm that produces schedules for different levels in large scale systems.

III. The Optimal Traffic-Division Arbitration Scheduler

The TDA scheduler algorithm with the optimized threshold can be described in the following steps:

1) Each user of the shared communication network is assigned a weight function.
2) An optimal threshold is then computed.
3) Dynamically choose the requests that their weights pass the threshold and give them an order which forms the transmission cycle.
4) Statically make the shared communication medium available to the chosen users according to the order that has been formed in Step 3.
5) Go to Step 1.

However we are introducing the TDA scheduler for standard NCS setup, we are looking beyond the NCS applications. Our goal is to introduce a general purpose scheduler that can be general enough to handle any process that includes scheduling and control. In fact, our goal is to provide solutions for optimal scheduling and control in fields like air traffic management, economic and medical systems. Therefore, in the following, we are using an air traffic management example to conceptualize the optimized algorithm of the TDA scheduler.

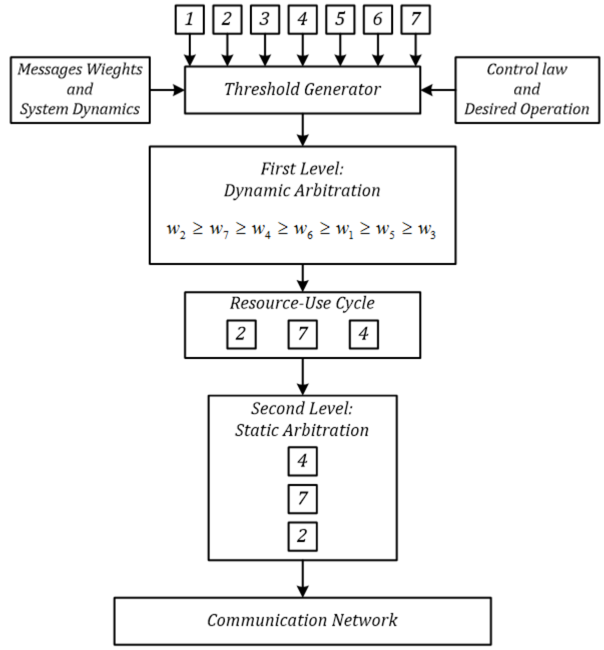

Fig. 1. An example to illustrate the optimized algorithm of the TDA scheduler.

Figure 1 explains our example. In the figure, there are 7 users that compete to gain the access to a shared resource which can be an airport or an air sector. First, each user is assigned a weight that represents the priority of the user's request. The weight can be a general function that represents any aspect of priority. The following is an example of a weight function that represents different priority aspects:

$$R_i(w_i) = R_i(w_i^p, w_i^c, w_i^d),$$

where,

- $i = 1, 2, \ldots, N$
- N: total number of the users of the shared resource,
- w_i^p, priority weight: the weight that represents the request order (e.g., the first flight that requests the resource use gets the highest priority and the last gets the lowest priority),
- w_i^c, category weight: each flight's type has different weight (e.g. passenger flight, commercial flight and political flight),
- w_i^d, delay weight: the weigh that represents the number of time delay units that the flight has passed through:
  1) For departure from an airport, the number of time delay units is directly related to the hold on time in the ground.
  2) For landing in an airport, the number of time

delay units is directly related to the hold on time in the air.

3) For entering a sector, the number of time delay units is directly related to the total delay which is the difference between the scheduled time of arrival and the estimated time of arrival.

The second stage in the scheduler algorithm is to generate the threshold based on the available information about the current state of the system, the control law and the weight functions of each user. The threshold then is used to formulate the Resource-Use Cycle (RUC) which is equivalent to the transmission cycle in the NCS setup. We assume that only the highest three weights of the competing users passed the threshold level, which are 2, 7 and 4 in the example. The formulation of the RUC represents the dynamical level of the scheduling arbitration. The final step of the scheduler algorithm is to statically make the resource available to the users that their weights passed to the static level. In this case, user 2 will be the first to use the shared resource of the current resource-use cycle then user 7 and 4 get the access respectively.

IV. Problem Formulation

In our previous work [8], we generated the threshold by a trial and error searching criterion in order to introduce and explain the algorithm of the TDA scheduler and its use in the NCS. In this paper, we aim to achieve two goals. The first objective is to find a systematic way to find the optimal threshold instead of the inefficient trial and error criterion, while the second objective is to extend the use of the TDA to be a general purpose scheduling protocol that can be used in wider range of applications other than the NCS.

In order to achieve the two goals that we mentioned above, we formulate an optimal control problem. We first formulate the cost function of the optimal control problem and define the constraints. Since our problem include dynamical terms (the system dynamics) and logical terms (the scheduling variables), then we need to introduce the Mixed Logical-Dynamical (MLD) framework to solve our optimization problem.

A. Problem Notations and Variables

One of the objectives of this task is to find a systematic optimal way to compute the threshold. In order to achieve this goal, we need to form an optimal control problem. The solution of this optimal control problem provides a dynamic optimal threshold that achieves a pre-described objective.

To form our optimal control-scheduling problem, let us define the following variables for each sampling period:

1) The arbitration threshold is $\gamma(k)$.
2) The index set of all network users $I_x = \{1, 2, \ldots, n\}$.
3) The weighted error of the message of the i^{th} user is $e_i(k)$.
4) The scheduling function that includes the access order for all the network users is $\delta(k)$.

To define the relation between the above variables, let us consider the following case. If the weighted error of the message of the i^{th} user exceeded the arbitration threshold, then $e_i \geq \gamma, \Rightarrow e_i - \gamma \geq 0$ where $i \in I_x$. According to the TDA algorithm, the message in this case passes to the next level of arbitration. For instance, let the number of all the networks users be $n = 3$, and if the error of all the competing messages exceeded the arbitration threshold, then $e_1 - \gamma \geq 0, e_2 - \gamma \geq 0$, and $e_3 - \gamma \geq 0$:

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} - \gamma \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Let

$$\delta_i = \begin{cases} 1 & \text{if } e_i - \gamma \geq 0 \\ 0 & \text{if } e_i - \gamma < 0, \end{cases}$$

and $\boldsymbol{\delta} = \begin{bmatrix} \delta_1 & \delta_2 & \delta_3 \end{bmatrix}^\top$.

The above formulation shows the relation between the arbitrating threshold γ , the access order δ_i and the weighted error e_i for each message.

B. Cost Function Construction

Consider the following dynamics of the system:

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k),$$

where $\boldsymbol{x}$ and $\boldsymbol{u}$ are the state and the input of the system respectively, then in the case of solving an optimal control problem using the linear quadratic regulator (LQR), the general form of the cost function and the constraints would be:

$$\min J = \boldsymbol{x}(N)^\top \boldsymbol{Q}_0 \boldsymbol{x}(N)$$

$$+ \sum_{k=0}^{N-1} \left[\boldsymbol{x}(k)^\top \boldsymbol{Q} \boldsymbol{x}(k) + \boldsymbol{u}(k)^\top \boldsymbol{R} \boldsymbol{u}(k) + \boldsymbol{\gamma}(k)^\top \boldsymbol{S} \boldsymbol{\gamma}(k) \right],$$

$$\text{subject to:} \begin{cases} \boldsymbol{x}(k+1) = \boldsymbol{A}x(k) + \boldsymbol{B}\boldsymbol{u}(k) \\ \sum_{i=1}^{m} \delta_i(k) = 1 \\ \delta_i(k) = 1 \Rightarrow e_i(k) - \gamma_i(k) \geq 0, \\ \delta_i(k) = 0 \Rightarrow u_i(k) = u_i(k-1) \end{cases} \quad (1)$$

where $\boldsymbol{Q}, \boldsymbol{R}$ and $\boldsymbol{S}$ are weighting factors for the optimal control problem and δ_i is the scheduling variable for $i \in I_x$.

We consider that the transmission channel can transmit only one message at any given time. This limitation of the transmission channel is represented in the constraint $\sum_{i=1}^{m} \delta_i(k) = 1$. In addition, if the value of the scheduling function that corresponds to the i^{th} user equals 1 ($\delta_i(k) = 1$), then the error of the competing message of the i^{th} user exceeded the arbitration threshold. This can be represented by the third constraint ($\delta_i(k) = 1 \Rightarrow e_i(k) - \gamma_i(k) \geq 0$). Also, if $\delta_i(k) = 0$, then the control law of the corresponding user will not be updated during this current sampling period. This case is represented by the

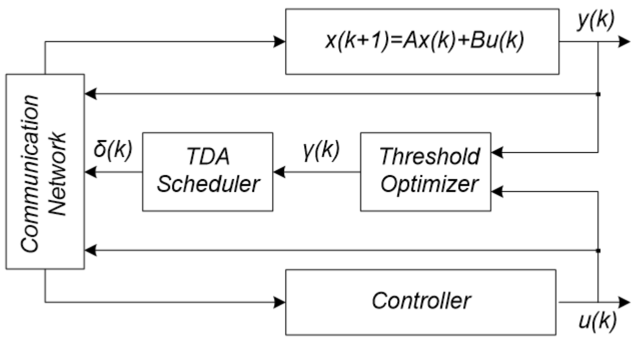

Fig. 2. A networked control system with mixed logical and dynamical inputs.

fourth constraint ($\delta_i(k) = 0 \Rightarrow u_i(k) = u_i(k-1)$).

Obviously, the optimal control problem that we are solving includes dynamic variables that are changing according to the system's dynamics, while at the same time the scheduling variables are logical. After we find the right formulation of the cost function and the system constraints, then we can find the optimal threshold γ. For the cost function in (1), the augmented cost function is:

$$J_a = \boldsymbol{x}(N)^\top \boldsymbol{Q}_0 \boldsymbol{x}(N)$$
$$+ \sum_{k=0}^{N-1} \{\boldsymbol{x}(k)^\top \boldsymbol{Q}\boldsymbol{x}(k) + \boldsymbol{u}(k)^\top \boldsymbol{R}\boldsymbol{u}(k) + \boldsymbol{\gamma}(k)^\top \boldsymbol{S}\boldsymbol{\gamma}(k)\}$$
$$+\boldsymbol{\lambda}^\top(k+1)[-\boldsymbol{x}(k+1) + \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k)] + \boldsymbol{\mu}^\top \boldsymbol{C}.$$

The optimality conditions can be written as:

$$\begin{cases} \frac{\partial J_a}{\partial \boldsymbol{u}(k)} = \boldsymbol{u}^\top(k)\boldsymbol{R} + \boldsymbol{\lambda}^\top(k+1)\boldsymbol{B} = \boldsymbol{0} \\ \frac{\partial J_a}{\partial \boldsymbol{\gamma}(k)} = \boldsymbol{\gamma}^\top \boldsymbol{S} = \boldsymbol{0} \\ \frac{\partial J_a}{\partial \boldsymbol{\lambda}(k+1)} = -\boldsymbol{x}(k+1) + \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) = \boldsymbol{0} \\ \frac{\partial J_a}{\partial \boldsymbol{x}(k)} = \boldsymbol{x}(k)^\top \boldsymbol{Q} - \boldsymbol{\lambda}^\top(k) + \boldsymbol{\lambda}^\top(k+1)\boldsymbol{A} = \boldsymbol{0} \\ \frac{\partial J_a}{\partial \boldsymbol{\mu}(k)} = \boldsymbol{C}(\boldsymbol{x}(k), \boldsymbol{u}(k), \boldsymbol{\gamma}(k), k, \text{Mixed Terms}) = \boldsymbol{0} \end{cases}$$

Having the term $\boldsymbol{\mu}^\top(k+1)\boldsymbol{C}$ that includes mixed logical and dynamical terms, the augmented cost function cannot be found. Therefore, using a different approach is necessary to find a solution for the optimal control problem. Accordingly, we need to use the mixed logical dynamical (MLD) optimization framework which is defined in [18].

C. Problem Formulation Conversion to the MLD Framework

The NCS shown in Figure 2, has a combination of input control terms, $\boldsymbol{u}$ and logical terms $\boldsymbol{\delta}$. For a given initial state $\boldsymbol{x}(0)$ and a given final time N, to find an optimal control sequence, $\boldsymbol{u}(N-1)$, and an optimal scheduling sequence $\boldsymbol{\delta}(N-1)$, we need to convert the logical constraints into normal inequality constraints. Then we need to replace the mixed terms with new linear terms. To achieve that, we need use the MLD framework. In order to manipulate the mixed constraints, we start with the first logical constraint of the optimization problem (1),

$$\delta_i(k) = 1 \Rightarrow e_i(k) - \gamma_i(k) \geq 0. \quad (2)$$

Define the dynamic variable d to be:

$$d_i(k) = e_i(k) - \gamma_i(k).$$

The value of d_i is subject to saturation constraints where the upper bound is U_i^d, and the lower bound is L_i^d, and both U_i^d and L_i^d are nonnegative values. Therefore,

$$\delta_i(k) = 1 \Rightarrow d_i(k) \geq 0 \Leftrightarrow \delta_i(k)d_i(k) \geq 0.$$

Let,

$$D_i(k) = \delta_i(k)d_i(k) \Rightarrow D_i(k) \geq 0.$$

D_i can expressed equivalently as in [18]:

$$\begin{aligned} D_i(k) &\leq U_i^d \delta_i(k) \\ D_i(k) &\geq L_i^d \delta_i(k) \\ D_i(k) &\leq d_i(k) - L_i^d(1 - \delta_i(k)) \\ D_i(k) &\geq d_i(k) - U_i^d(1 - \delta_i(k)). \end{aligned}$$

Now we need to use the same transformation techniques to manipulate the second logical constraint in the optimization problem (1):

$$\delta_i(k) = 0 \Rightarrow u_i(k) = u_i(k-1). \quad (3)$$

The following is an equivalent expression for (3) that can be used to replace the use of the connective operator ($\Rightarrow$).

$$u_i(k) - u_i(k-1) = \delta_i(k)u_i(k) - \delta_i(k)u_i(k-1). \quad (4)$$

We can see that (4) has mixed logical-dynamical terms. Therefore we need to introduce new variables as follows:

$$\begin{aligned} z_i(k) &= \delta_i(k)u_i(k) \\ w_i(k) &= \delta_i(k)u_i(k-1). \end{aligned}$$

Then using the transferring technique as in [19], we get

$$\begin{cases} z_i(k) \leq U_i^z \delta_i(k) \\ z_i(k) \geq L_i^z \delta_i(k) \\ z_i(k) \leq u_i(k) - L_i^z(1 - \delta_i(k)) \\ z_i(k) \geq u_i(k) - U_i^z(1 - \delta_i(k)) \\ w_i(k) \leq U_i^w \delta_i(k) \\ w_i(k) \geq L_i^w \delta_i(k) \\ w_i(k) \geq u_i(k) - L_i^w(1 - \delta_i(k)) \\ w_i(k) \geq u_i(k) - U_i^w(1 - \delta_i(k)), \end{cases}$$

The value of z_i and w_i is subject to saturation constraints where the upper bounds of z_i and w_i are U_i^z and U_i^w respectively, and the lower bounds are L_i^z and L_i^w. In addition, all these upper and lower bounds are nonnegative values. Putting the mixed constraints in (2) and (3) all together, we get:

$$\delta_i(k)=0 \Rightarrow u_i(k)=u_i(k-1) \equiv$$
$$\begin{cases} z_i(k) \le U_i^z \delta_i(k) \\ z_i(k) \ge L_i^z \delta_i(k) \\ z_i(k) \le u_i(k) - L_i^z(1-\delta_i(k)) \\ z_i(k) \ge u_i(k) - U_i^z(1-\delta_i(k)) \\ w_i(k) \le U_i^w \delta_i(k) \\ w_i(k) \ge L_i^w \delta_i(k) \\ w_i(k) \ge u_i(k) - L_i^w(1-\delta_i(k)) \\ w_i(k) \ge u_i(k) - U_i^w(1-\delta_i(k)), \end{cases}$$

and

$$\delta_i(k)=1 \Rightarrow e_i(k)-\gamma_i(k) \ge 0 \equiv$$
$$\begin{cases} D_i(k) \le U_i^D \delta_i(k) \\ D_i(k) \ge L_i^D \delta_i(k) \\ D_i(k) \le d_i(k) - L_i^D(1-\delta_i(k)) \\ D_i(k) \ge d_i(k) - U_i^D(1-\delta_i(k)), \end{cases}$$

If we define the following new variables:

$$\boldsymbol{U}_{DD} = \mathrm{diag}(U_1^d,\ldots,U_n^d),\ \boldsymbol{L}_{DD} = \mathrm{diag}(L_1^d,\ldots,L_n^d)$$
$$\boldsymbol{U}_{ZD} = \mathrm{diag}(U_1^z,\ldots,U_n^z),\ \boldsymbol{L}_{ZD} = \mathrm{diag}(L_1^z,\ldots,L_n^z)$$
$$\boldsymbol{U}_{WD} = \mathrm{diag}(U_1^w,\ldots,U_n^w),\ \boldsymbol{L}_{WD} = \mathrm{diag}(L_1^w,\ldots,L_w^z)$$
$$\boldsymbol{\Delta} = \left[\delta(0)^\top\ \ldots\ \delta(N-1)^\top\right]^\top,$$
$$\boldsymbol{u} = \left[u(0)^\top\ \ldots\ u(N-1)^\top\right]^\top,$$

and $\boldsymbol{\Delta}^c$ is the complement of $\boldsymbol{\Delta}$. We can formulate the problem in a compact form and then we can convert the MLD form to the quadratic programming form. The general form of the quadratic programming is as follows:

$$\begin{aligned} \min\ & \tfrac{1}{2}\boldsymbol{v}^\top \boldsymbol{H}\boldsymbol{v} + \boldsymbol{f}^\top \boldsymbol{v} \\ \text{s.t.}\ & \boldsymbol{A}\boldsymbol{v} \le \boldsymbol{B}, \end{aligned}$$

where,

$$\boldsymbol{v} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{u} \\ \boldsymbol{\Gamma} \\ \boldsymbol{z} \\ \boldsymbol{w} \\ \boldsymbol{e} \end{bmatrix}, \boldsymbol{f} = \begin{bmatrix} 0 \\ \boldsymbol{Q}_0\boldsymbol{x}(N) \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \boldsymbol{H} = \begin{bmatrix} 2\boldsymbol{Q} & & & \boldsymbol{0} \\ & 2\boldsymbol{R} & & \\ & & 2\boldsymbol{S} & \\ \boldsymbol{0} & & & \ddots & \\ & & & & \boldsymbol{O} \end{bmatrix},$$

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{I}_z & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & -\boldsymbol{I}_z & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & -\boldsymbol{I}_u & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{I}_z & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & -\boldsymbol{I}_u & \boldsymbol{O} & \boldsymbol{O} & -\boldsymbol{I}_z & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{I}_w & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & -\boldsymbol{I}_w & \boldsymbol{O} \\ \boldsymbol{O} & -\boldsymbol{I}_u & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{I}_w & \boldsymbol{O} \\ \boldsymbol{O} & -\boldsymbol{I}_u & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & -\boldsymbol{I}_w & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{I}_e \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & -\boldsymbol{I}_e \\ \boldsymbol{O} & -\boldsymbol{I}_u & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{I}_e \\ \boldsymbol{O} & -\boldsymbol{I}_u & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & -\boldsymbol{I}_e \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} \boldsymbol{U}_{ZD}\boldsymbol{\Delta} \\ \boldsymbol{L}_{ZD}\boldsymbol{\Delta} \\ -\boldsymbol{L}_{ZD}\boldsymbol{\Delta}^c \\ -\boldsymbol{U}_{ZD}\boldsymbol{\Delta}^c \\ \boldsymbol{U}_{WD}\boldsymbol{\Delta} \\ \boldsymbol{L}_{WD}\boldsymbol{\Delta} \\ -\boldsymbol{L}_{WD}\boldsymbol{\Delta}^c \\ -\boldsymbol{U}_{WD}\boldsymbol{\Delta}^c \\ \boldsymbol{U}_{DD}\boldsymbol{\Delta} \\ \boldsymbol{L}_{DD}\boldsymbol{\Delta} \\ -\boldsymbol{U}_{DD}\boldsymbol{\Delta}^c \\ -\boldsymbol{L}_{DD}\boldsymbol{\Delta}^c \end{bmatrix}.$$

The vector $\boldsymbol{v}$ is a concatenation of the vectors $\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\Gamma}, \boldsymbol{z}, \boldsymbol{w}$ and $\boldsymbol{e}$. Since this optimization is performed dynamically at every time step in the networked control model, the sub-vector $\boldsymbol{x}$ has two elements, $\boldsymbol{x}(N-1)$ and $\boldsymbol{x}(N)$. The vector containing $\boldsymbol{u}, \boldsymbol{\Gamma}, \boldsymbol{z}, \boldsymbol{w}$ and $\boldsymbol{e}$ contains single elements at step $N-1$. The vector $\boldsymbol{f}$ contains the terms for the final cost. Here, we assume that the final cost terms are zero. The matrix $\boldsymbol{H}$, is a diagonal matrix containing the terms of the LQR cost function. Since the control terms $\boldsymbol{x}, \boldsymbol{u}$ and $\boldsymbol{\Gamma}$ are the only ones being optimized, and the bound terms $\boldsymbol{z}, \boldsymbol{w}$ and $\boldsymbol{e}$ were not included in the LQR, hence their corresponding terms along the diagonal are zero. The matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ corresponded to the bounded constraints.

At every step, the optimal $\boldsymbol{v}^*$ is calculated and the optimal threshold $\boldsymbol{\Gamma}^*$ terms are extracted and then forwarded to the scheduling routine to enforce the optimal access order to use the shared resource.

V. Numerical Results

In this section we introduce numerical examples and show simulation results of the optimized TDA scheduler and compare its behavior with the round-robin scheduling criterion. We chose the Round-Robin among all other schedulers because of its wide use in most scheduling algorithms in a lot of fields and applications other than NCS. As we mentioned in the introduction section, one of our goals is to extend the use of the TDA scheduling algorithm to other fields other than NCS in order to make it a general purpose scheduler.

In the following we are introducing a numerical example where we calculate the average time delay of all the resource users in each time step when we use the TDA as a scheduler. Then we calculate the average time delay of all resource users when we use the round-robin as a scheduler in order to compare the two averages and evaluate the proposed optimized scheduler. In addition, we do the same comparison using the standard deviation of the average time delay.

The LTI MIMO example used in the simulations is adopted from [20]:

$$\boldsymbol{A} = \begin{bmatrix} 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \\ 2&2&0&0 \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} 0&1&0&0 \\ 0&0&0&1 \\ 1&0.2&-1&-2 \\ 0&0.5&0&0.5 \end{bmatrix} \boldsymbol{C} = \begin{bmatrix} 1&0&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \end{bmatrix}.$$

It is an unstable system with two controllers. In the controller design, only the local information are available. In our simulation of this system the feedback-loops are closed though communication network. In every time step of the simulation we compute the optimal threshold and the optimal control law. In order to compare the optimized

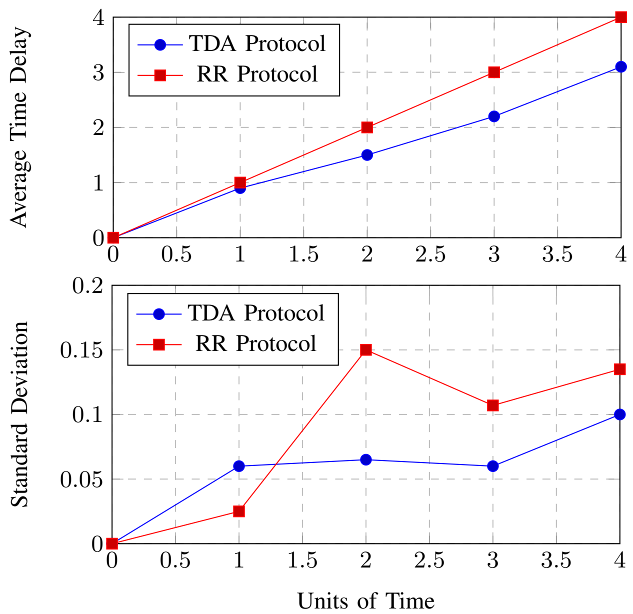

Fig. 3. Comparison between the time delay of TDA and RR for example 1 (4 units of time).

TDA scheduler with the Round-Robin static scheduler, we compute the time delay for each node which is the amount of time elapsed from the time it attempted to use the resource to the time it granted the use of it. We accumulate the delay for each attempt for each node though out the running time of the simulation. Then, we take the average accumulated time delay for all resource users every predetermined time interval. In addition, for each time interval we compute the standard deviation of the average accumulated time delay.

Figure 3 shows the average time delay and the standard deviation for the TDA and the RR schedulers. In this run of the simulation we considered 4 units of time. It is clear that the average accumulated time delay for each node under the use of the optimized TDA scheduler is less than the average accumulated time delay of the RR scheduler.

VI. Summary and Conclusions

The network-induced time delay is one of the major challenges that appear when we need to deal with the design or the analysis of any Networked Control System (NCS). The delay can be tackled in the network level by performing control of network actions. Since the process of prioritizing the network access is the main contributor of the network-induced time delay, then the efficient way to control the time delay is to improve the scheduling criterion. In this paper we propose an optimal scheduling criterion that improves the quality of service for each node in the network and at the same time maintains the stability of the control system. The proposed optimal scheduling protocol is a hybrid scheduling criterion that combines the advantages of the dynamic and the static scheduling protocols. First, the algorithm of the optimal scheduling is discussed. Second, the optimized control-scheduling problem is formulated. The mixed logical-dynamical optimization problem is then solved and a numerical examples is presented.

References

[1] CAN specification version 2.0. *Robert Bosch GmbH*, Stuttgart, Germany, 1991.

[2] U. Kiencke, S. Dais and M. Litschel, "Automotive serial controller area networ," *In SAE International Congress No. 860391*, Detroit, MI, 1986.

[3] T.-C. Yang, "Networked control system: a brief survey," Control Theory and Applications, *Proceedings of IEEE Conference on Control Theory Applications*, vol. 153, no. 4, pp.403–412, 10 July 2006.

[4] H. F. Othman; Y. R. Aji, F. T Fakhreddin, and A.R. Al-Ali, "Controller Area Networks: Evolution and Applications," *Information and Communication Technologies, ICTTA '06*, vol. 2, pp. 3088–3093, 2006.

[5] H. M. Newman, "Integrating building automation and control products using the BACnet protocol," *ASHRAE Journal*, vol. 38, no. 11, pp. 36–42, November, 1996.

[6] G. C. Walsh and Y. Hong, "Scheduling of networked control systems," *IEEE Control Systems magazine*, vol. 21, no. 1, pp. 57–65, February 2001.

[7] G. C. Walsh, Y. Hong and L. Bushnell, "Stability analysis of networked control systems," *Proceedings of the 1999 American Control Conference*, vol. 4, pp.2876–2880, 1999.

[8] A. Elmahdi, A. F. Taha, S. Hui and S. H. Żak, "A hybrid scheduling protocol to improve quality of service in networked control systems," *2012 50th Annual Allerton Conference on Communication, Control, and Computing*, pp. 98–105, 1–5 October 2012.

[9] R. Brockett, "Stabilization of motor networks," *in Proc. of IEEE Conf. Decision and Control*, pp. 1484–1488, New Orleans, LA, December 1995.

[10] D. Nešić and A. R. Teel, "Inputoutput stability properties of networked control systems," *IEEE Transactions on Automatic Control*, vol. 49, pp. 1650–1667, 2004.

[11] D. Nešić and A. R. Teel,, "Input-to-state stability of networked control system," *Automatica*, vol. 40, no. 12, pp. 2121–2128, December 2004.

[12] M. Tabbara and D. Nešić, "InputOutput Stability of Networked Control Systems With Stochastic Protocols and Channels," *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1160–1175, June 2008.

[13] M. Tabbara, D. Nešić and A. R. Teel, "Inputoutput stability of wireless networked control systems," *In Proceedings of 44th IEEE Conference on Decision and Control*, pp. 209-214, Seville, Spain, 2005.

[14] M. Tabbara, D. Nešić and A. R. Teel, "Stability of Wireless and Wireline Networked Control Systems," *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1615–1630, September 2007.

[15] Y. Kim, H. Park and W. Kwon, "A scheduling method for network-based control systems," *Proceedings of the 1998 American Control Conference*, vol. 2, pp. 718–722, June 21–26, 1998.

[16] H. Park, Y. Kim, D. Kim and W. Kwon, "A scheduling method for network-based control systems," *IEEE Transactions on Control Systems Technology* , vol. 10, no. 3, pp. 318–330, May 2002.

[17] D. Kim, Y. Lee, W. Kwon and H. Park, "Maximum allowable delay bounds of networked control systems," *Control Eng. Pract.*, vol. 11, no. 11, pp.'13011313, November 2003.

[18] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407-427, 1999.

[19] E. M. Gaid, M. Cela and A. Hamam, "Optimal Integrated Control and Scheduling of Systems with Communication Constraints," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 4, pp. 776–787, July 2006.

[20] A. Nguyen, Q. P. Ha, S. Huang, and H. Trinh, "Observer-based decentralized approach to robotic formation control," Proceedings of the 2004 Australian Conference of Robotics and Automation, pp. 1-8, Australia, December 2004.

[21] Q. Ha and H. Trinh, "Observer-based control of multi-agent systems under decentralized information structure," Int. J. Sys. Science, Vol. 35, No. 12, pp.719–728, October 2004.