Steady State Performance Analysis of Multiple State-based Schedulers with CSMA

Chithrupa Ramesh, Henrik Sandberg, and Karl Henrik Johansson

Abstract— **In this paper, we analyze the performance of multiple event-based systems that share access to the same network. Transmissions are attempted only when a local state-based scheduler generates an event, and access to the network is determined using a Carrier Sensing Multiple Access (CSMA) protocol. In general, the interactions in such a multiple access network introduce correlations between the system variables of the various loops, and the respective traffic contributions as well. Hence, analyzing the performance of this network is difficult. However, a class of state-based schedulers, introduced in the paper, permits a joint analysis of the scheduler and the Contention Resolution Mechanism (CRM). The analysis is based on a Markov model, which is validated through simulations. The resulting steady-state model makes it possible to characterize the statistics of packet arrivals in this network.**

I. INTRODUCTION

In networked control systems, the closed loop is operated with a network between the sensor and the controller, or the controller and the actuator, or both. We consider a network of M such closed loop systems, as shown in Fig. 1, where the sensors and controllers communicate over a shared network, but the controllers and actuators use a dedicated link each. This is typical of many industrial settings where actuators require power lines to be drawn, and can be simultaneously connected by wires [1]. The shared sensing link does not permit simultaneous transmissions, and there is much benefit in reducing the data generated by the sensors [2], [3].

To accomplish this, each sensor could operate along with a local scheduler, where the scheduling criterion or event triggering policy decides when to schedule a sample for transmission over the network [4], [5], [6]. In this paper, such a scheduler is called a state-based scheduler as it results in scheduling decisions which are correlated to the state of the plant. We define these state-based schedulers for discrete time linear stochastic plants, and use a scheduling policy which acts on an information pattern. The information pattern is a sufficient statistic for the state, and permits a general formulation for such a system.

The shared network still requires a Medium Access Controller (MAC) to arbitrate access to the medium. We consider MACs which use a contention-based access protocol, such as CSMA, as against a contention-free protocol, such as Time Division Multiple Access (TDMA) [7]. CSMA protocols do not guarantee a transmission, but use a CRM which may result in collisions [8]. Despite this, we consider CSMA protocols for event-based systems as they do not require a centralized scheduler, and are implemented in a distributed manner among the nodes. Their ad hoc mode of operation is well suited to events, which are aperiodically generated by the nodes, and hence difficult to accommodate in a schedule.

This work was supported by the Swedish Research Council, VINNOVA (The Swedish Governmental Agency for Innovation Systems), the Swedish Foundation for Strategic Research, the Knut and Alice Wallenberg Foundation and the EU project FeedNetBack.

C. Ramesh, H. Sandberg and K.H. Johansson are with the ACCESS Linnaeus Centre, Electrical Engineering, Royal Institute of Technology (KTH), Stockholm, Sweden. {cramesh,hsan,kallej}@ee.kth.se

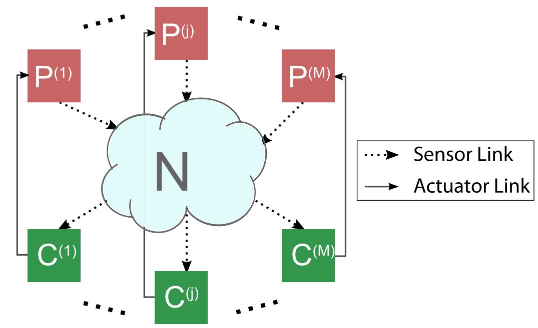

Fig. 1. M closed loops, consisting of a plant ($\mathcal{P}$) and a controller ($\mathcal{C}$) each, use a shared network ($\mathcal{N}$) for communication between the respective sensors and controllers. Note that the controllers and actuators communicate over a dedicated point to point link, which suffers no losses.

Now, we are interested in choosing parameters of the state-based scheduler. This requires us to understand how these parameters affect the performance of the closed loop system. However, analyzing the performance of such a network is not an easy task as the CRM introduces correlations between data packets from different control loops. Even characterizing the traffic on such networks is not easy. The state-based scheduler adapts the event generation, or its traffic contribution, to the transmission history, through the varying statistics of the input arguments to the scheduling policy. However, the transmission history in turn depends on the CRM's response to the traffic in the network. Thus, a joint analysis of the state-based scheduler and the CRM is required.

The joint analysis is the main contribution of this work. We use a Markov chain to describe a class of scheduling policies and the CRM, as it permits us to recover an independence property at the CRM level, which facilitates the steady state analysis. This approach is based on Bianchi's analysis of the Distributed Coordination Function in 802.11 [9]. The key assumption in this model is the same as in Bianchi's; we assume that the conditional probability of a busy channel for a node that attempts to transmit, is given by an independent probability p for each node. In other words, we assume that nodes attempting to transmit always see the network in steady state. Once we define our Markov chain, we can solve to find p. The key assumption of conditional independence is verified through simulations and is indeed found to hold quite well even under unsaturated traffic conditions.

Some of the earlier work in this area includes a partial analysis of event triggered nodes with CSMA/CA [10]. This work highlighted the difficulties in analyzing such a network. A full analysis with Aloha was presented in [11], which assumed independent packet losses. A simple steady state model was presented in [12], but with an idealized CRM which results in no collisions. More recently, event-based systems which use Aloha and Slotted Aloha have been analyzed in [13], but with an event-triggering law that has been chosen to result in independent packets. The work presented in this paper highlights the need for a joint analysis between the CRM and the event-triggering policy.

The rest of the paper is organized as follows. We present the network setup in Section II and derive properties needed for the Markov description in Section III. We delve into the Markov chain based description in Section IV and solve to obtain a simple model. Finally, we present some simulation results which confirm our assumptions in Section V.

II. PRELIMINARIES

We consider a network of M plants and controllers (indexed by $j \in \{1, \ldots, M\}$), which communicate over a shared channel with a state-based scheduler in the loop, as shown in Fig. 2. The different blocks in the figure are explained below.

Plant: The plant $\mathcal{P}^{(j)}$ has state dynamics given by

$$x_{k+1}^{(j)} = A_j x_k^{(j)} + B_j u_k^{(j)} + w_k^{(j)} , \tag{1}$$

where $A_j \in \mathbb{R}^{n\times n}$, $B_j \in \mathbb{R}^{n\times m}$ and the initial state $x_0^{(j)}$ and the process noise $w_k^{(j)}$ are i.i.d. zero-mean Gaussians with covariance matrices $R_0^{(j)}$ and $R_w^{(j)}$, respectively. They are independent and uncorrelated to $x_0^{(i)}$ and $w_k^{(i)}$ respectively, for any other plant i in the network. This discrete time model is defined with respect to a sampling period T for each plant, and the sampling instants are generated by a synchronized network clock.

State Based Scheduler: There is a local scheduler $\mathcal{S}^{(j)}$, situated in the sensor node, between the plant and the controller, which decides if the state is to be sent across the network or not. The scheduler output $\gamma_k^{(j)} \in \{0, 1\}$, and is equal to 1 when the state $x_k^{(j)}$ is scheduled to be sent. The scheduling criterion is denoted by the policy $f^{(j)}$, which is defined on the information pattern of the scheduler $\mathbb{I}_k^{S^{(j)}}$ and is given by

$$\gamma_k^{(j)} = f_k^{(j)}(\mathbb{I}_k^{S^{(j)}}) , \tag{2}$$

where, $\mathbb{I}_k^{S^{(j)}} = \left[\{\boldsymbol{x}^{(j)}\}_0^k, \{\boldsymbol{y}^{(j)}\}_0^{k-1}, \{\boldsymbol{\gamma}^{(j)}\}_0^{k-1}, \{\boldsymbol{\delta}^{(j)}\}_0^{k-1}, \{\boldsymbol{u}^{(j)}\}_0^{k-1}\right]$, and δ_k is the MAC output given in (4). The bold font $\{\boldsymbol{a}^{(j)}\}_t^T$ denotes the set $\{a_t^{(j)}, a_{t+1}^{(j)}, \ldots, a_T^{(j)}\}$.

CRM: The CRM resolves contention between simultaneous channel access requests. The output of the CRM is denoted $\alpha_k^{(j)} \in \{0, 1\}$, and is given by

$$\alpha_k^{(j)} = \mathcal{R}(\gamma_k^{(j)}, \varrho) , \tag{3}$$

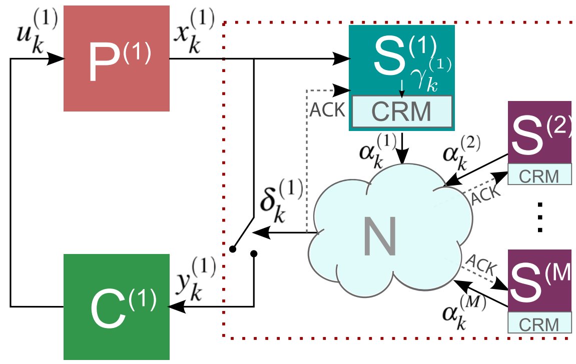

Fig. 2. A network ($\mathcal{N}$) of plants ($\mathcal{P}^{(j)}$), state-based schedulers ($\mathcal{S}^{(j)}$) and controllers ($\mathcal{C}^{(j)}$), for $j \in \{1, \ldots, M\}$, with an explicit CRM in the MAC. The variables γ, α and δ denote scheduler requests, CRM transmission attempts and the MAC output, respectively. The control loops {2,...,M} are represented with their schedulers alone.

where, ϱ denotes the set of parameters that characterize the CRM. The variable $\alpha_k^{(j)} = 1$ when the CRM permits the packet to access the network. The CRM used in some of the following sections is p-persistent CSMA with no retransmissions. In this CRM, a node which finds the channel busy persists to wait and transmits as soon as the channel becomes idle with the persistence probability p_α [8]. Thus, $\varrho = \{p_\alpha\}$ for this CRM.
Now, the MAC output $\delta_k^{(j)}$, is given by

$$\begin{aligned}\delta_k^{(j)} &= \mathcal{N}(\{\boldsymbol{\alpha}_k^{(j)}\}_{j=1}^M) \\ &= \begin{cases} 1 & \alpha_k^{(j)} = 1, \alpha_k^{(i)} = 0,\ i \neq j \\ 0 & \begin{cases} \alpha_k^{(j)} = 1, \sum_{i\neq j, i=1}^M \alpha_k^{(i)} > 0 \\ \alpha_k^{(j)} = 0 \end{cases} \end{cases} ,\end{aligned} \tag{4}$$

where, $(\delta_k^{(j)} = 1)$ indicates that a successful transmission occurs only when the CRM permits a transmission *and* none of the other nodes attempt to access the network.

Controller: The controller $\mathcal{C}^{(j)}$ receives $y_k^{(j)} = \delta_k^{(j)} x_k^{(j)}$ from the network, and the control law g is given by

$$u_k^{(j)} = g_k^{(j)}(\mathbb{I}_k^{C^{(j)}});\ \mathbb{I}_k^{C^{(j)}} = \left[\{\boldsymbol{y}^{(j)}\}_0^k, \{\boldsymbol{\delta}^{(j)}\}_0^k, \{\boldsymbol{u}^{(j)}\}_0^{k-1}\right] , \tag{5}$$

where $\mathbb{I}_k^{C^{(j)}}$ is the information available to the controller.

We are interested in analyzing the performance of a closed loop system in this network, or in particular, evaluating the probability of a successful transmission, i.e., $\Pr(\delta_k^{(j)} = 1)$. This is required to evaluate the closed loop system performance [14].

III. MARKOVIAN REPRESENTATION OF EVENTS

We reaffirm that the multiple access network introduces correlations between different control loops with state-based schedulers, as has been noted earlier in [10], [11]. We also show that in general, the events from a state-based scheduler need not possess a Markov property. Then, we identify a class of state-based schedulers with the Markov property.

Lemma 1: For the closed loop system with the plant (1), state-based scheduler (2), MAC output (4) and controller (5), the state $x_k^{(j)}$ becomes correlated to the states $x_k^{(i)}$ for $i \neq j, i \in \{1, \ldots, M\}$.

Proof: The state of any plant in the network $x_{k+1}^{(j)}$ is a function of the control signal $u_k^{(j)}$, from (1). The control signal is a function of the MAC output $\delta_k^{(j)}$ from (5). The MAC output is a function of the previous states $x_k^{(i)}$ of all the plants in the network, i.e., for $i = \{1, \ldots, M\}$, through the scheduler outcomes $\gamma_k^{(i)}$, as given by (2–4). This is true for every plant in the network, and thus, all the states are correlated. ■

In the following lemma, we establish that in general, events described by the state-based scheduler in (2) depend on the entire transmission history, and thus, do not possess a Markov property.

Lemma 2: For the closed loop system with the plant (1), state-based scheduler (2), MAC output (4) and controller (5), the probability of a packet being scheduled for transmission, i.e., $\Pr(\gamma_k^{(j)} = 1)$, is a function of the transmission history $\{\boldsymbol{\delta}^{(j)}\}_0^{k-1}$, for $j \in \{1, \ldots, M\}$.

Proof: Since this proof holds for every plant in the network, we drop the superscript j. The state can be written as the sum of two components: a forced response and a natural response, as shown in

$$x_n = \sum_{\ell=1}^{n} A^{\ell-1} B u_{n-\ell} + A^n x_0 + \sum_{r=1}^{n} A^{r-1} w_{n-r}$$

where $u_{n-\ell} = g_{n-\ell}(\mathbb{I}_{n-\ell}^C)$, $\gamma_n = f_n(\mathbb{I}_n^S)$, and the information sets can be rewritten as $\mathbb{I}_{n-\ell}^C = \{\boldsymbol{y}_0^{n-\ell}, \boldsymbol{\delta}_0^{n-\ell}, g_0^{n-\ell-1}\}$ and $\mathbb{I}_n^S = \{\boldsymbol{x}_0^n, f_0^{n-1}, \boldsymbol{\delta}_0^{n-1}, g_0^{n-1}\}$. The set of primitive random variables in the above expressions are $\omega = \{x_0, \boldsymbol{w}_0^{k-1}, \boldsymbol{\delta}_0^{k-1}\}$, where δ captures the effect of all the unknown variables from the rest of the network. Of these, the distributions of x_0 and $\boldsymbol{w}_0^{k-1}$ are known, along with the scheduler and controller policies $\{f, g\}$. Then, the transmission history $\boldsymbol{\delta}_0^{k-1}$ must be characterized to evaluate the probability of satisfying the scheduling criterion in (2). ■

Now, consider the class of schedulers given by

$$\gamma_k^{(j)} = \tilde{f}_k^j(\mathfrak{I}_k^{S(j)}); \quad \mathfrak{I}_k^{S(j)} = \left[x_0^{(j)}, \{\boldsymbol{w}^{(j)}\}_0^k, d_{k-1}^{(j)}\right] \qquad (6)$$

where d_k is the delay since the last transmitted packet at time k, as illustrated in Fig. 3. Specifically, $d_k = k - \tau_k$, where τ_k is the time index of the last transmitted packet, i.e. $\tau_k = \{\max\{n, -1\} : \delta_n = 1, n \leq k\}$.

Theorem 3: For the closed loop system with the plant (1), state-based scheduler (6), MAC output (4) and controller (5), the probability of a packet being scheduled for transmission, i.e., $\Pr(\gamma_k^{(j)} = 1)$, is a function of the delay since the last successfully transmitted packet $d_{k-1}^{(j)}$, for $j \in \{1, \ldots, M\}$.

Proof: Again, we drop the superscript j in this proof. The input arguments to the scheduler in (6) contain the initial state and process noise, which can be characterized without the transmission history. In other words, their distributions are known a priori. Then, only the delay since the last successfully transmitted packet $d_{k-1}^{(j)}$ is required to be known or characterized to evaluate the probability of satisfying a scheduling criterion such as (6). ■

Thus, by removing the transmission history, $\boldsymbol{\delta}_0^{k-1}$, and the past applied controls, $\boldsymbol{u}_0^{k-1}$, it is possible to characterize the probability of satisfying a scheduling criterion such as (6), given just the delay since the last transmitted packet. Note that the probabilities in Lemma 2 are often hard to compute. However, these probabilities are easier to characterize for the class of schedulers given by (6), as noted in the next section. Also, with Theorem 3, $d_{k-1}^{(j)})$ is a sufficient statistic for the transmission history $\{\boldsymbol{\delta}^{(j)}\}_0^{k-1}$. For this class of schedulers, we describe the scheduling policy and the CRM with a Markov model, in the next section.

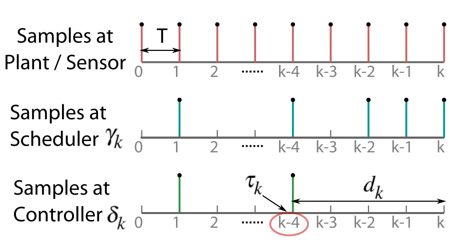

Fig. 3. An illustration of the delay since the last received packet, d_k, and the index of the last received packet, τ_k. The sensor samples the plant every T seconds, the scheduler chooses a few samples to be events ($\gamma_k = 1$), and some are successfully transmitted ($\delta_k = 1$).

IV. JOINT PERFORMANCE ANALYSIS

This section contains the main result of the paper, which is a joint model of the state-based scheduler and the CRM. We use the result from Theorem 3 to derive a Markov model and analyze the performance of the entire system. We also outline a few extensions to other network configurations.

A. A Simplified Model

Our modelling simplifications are described below. We assume that all plants in the network are sampled with the same period T, with a synchronized network clock, and that the CRM used is p-persistent CSMA with no retransmissions. We only consider scheduling policies such as in (6), and restrict the memory of the scheduler to simplify the modelling. We assume that the control data requires packet lengths which are small compared to the sampling period. However, note that the Markov model presented below exists, as shown in Theorem 3. For brevity, we skip the superscript (j), though the following description is for the j^{th} plant and scheduler over the network, where $j \in \{1, .., M\}$.

In the Markov chain in Fig. 4, we assign two indices, (s, d), to each state and denote the probability of being in the state with $p_{(s,d)}$. The index d represents the memory of the scheduler and is equal to the delay since the last successful transmission when $d < F$, where F is the finite memory of the scheduler. Note that restricting the memory of the scheduler does not constrain the burst length due to the infinite looping allowed in the last stage of the scheduling policy. The states (s, d) and $(s, d{+}1)$ are one sampling period away from each other. The index s represents the four states a packet can be in during a sampling period. These are

1) $s = -1$ [Idle State]: All the other states return to this state before the next sampling instant, when the next transition occurs. The initial state $(-1, 0)$ indicates the

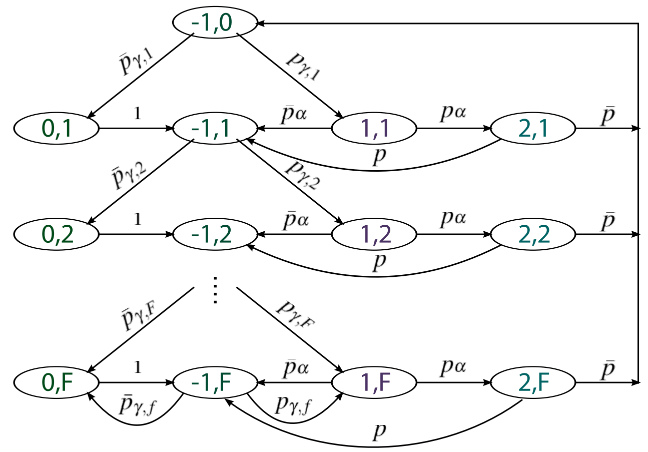

Fig. 4. A Markov chain representation for a state-based scheduler and a simple CRM such as p-persistent CSMA with no retransmissions. The variable F denotes the limit on the memory of transmission history used by the scheduler.

idle state before the next sampling instant following a successful transmission.

2) $s = 0$ [Non-scheduled State]: This state is reached when the scheduler output $\gamma_k = 0$. A transition out of this state occurs instantaneously, and always to the idle state $(-1, d)$ to wait for the next sampling instant.
3) $s = 1$ [Scheduled State]: This state is reached when $\gamma_k = 1$. A transition out of this state occurs within one sampling period to eventually reach an idle state, possibly through a transmission state.
4) $s = 2$ [Transmission State]: This state is reached when the contention resolution mechanism permits a channel access, or when $\alpha_k = 1$ in Fig. 2. Only a node in this state actually attempts a transmission. A transition out of this state occurs instantaneously, with two possibilities: transmission success or failure.

The transition probabilities in Fig. 4 are explained below:

- $p_{\gamma,d}$ denotes the probability of being scheduled at a sampling instant, thus marking a transition from state $(-1, d-1)$ to $(1, d)$. Conversely, $\bar{p}_{\gamma,d} = 1-p_{\gamma,d}$ denotes the probability of not being scheduled or transitioning from state $(-1, d-1)$ to $(0, d)$.
- $p_{\gamma,f}$ denotes the probability of being scheduled in the terminal stage of the scheduler.
- p_α denotes the probability of accessing the channel through the CRM, and marks a transition from state $(1, d)$ to $(2, d)$. Conversely, $\bar{p}_\alpha = 1 - p_\alpha$ represents the probability of returning to the idle state without attempting a transmission.
- p denotes the conditional probability of a busy channel, conditioned on nodes that attempt to transmit in state $(2, d)$. The node then reverts to the idle state $(-1, d)$. Otherwise, the transmission is successful and the node reverts to the initial state $(-1, 0)$.

The scheduler probabilities $p_{\gamma,d}$ and $p_{\gamma,f}$ can be computed along with the steady state probability p, as explained at the end of Section IV-B. The channel access probability p_α is provided for a given MAC.

B. Steady State Performance Analysis

For the Markov model described above, we can state the following result.

Theorem 4: For the closed loop system with the plant (1), state-based scheduler (6), MAC output (4) and controller (5), the probability of a successful transmission in steady state is given by

$$\Pr(\delta_k^{(j)} = 1) = (1 - p^{(j)}) \cdot p_{TX}^{(j)} \ , \tag{7}$$

where, $p^{(j)}$ is the conditional probability of a busy channel for nodes attempting to transmit (11), and $p_{TX}^{(j)}$ is the probability that a node attempts to transmit, or is in the $(2, d)$ states (9).

Proof: We begin by evaluating the probabilities $p_{(s,d)}$, in steady state, using the transition probabilities defined above, and relate the Markov models of all the nodes in the network to find an expression for the probability of a successful transmission.

The state $(-1, d)$ is always reached unless there is a successful transmission. The probability of a successful transmission in the d^{th} stage is given by $p_{\gamma,d}p_\alpha\bar{p}$, which gives us the recursive expression in

$$\begin{aligned} p_{(-1,d)} &= (1 - p_{\gamma,d}p_\alpha\bar{p})p_{(-1,d-1)} \ , d = \{1, \ldots, F-1\} \ , \\ p_{(-1,F)} &= \frac{1 - p_{\gamma,d}p_\alpha\bar{p}}{p_{\gamma,f}p_\alpha\bar{p}} p_{(-1,F-1)} \ . \end{aligned} \tag{8}$$

In the final stage, $(-1, F)$ can be reached in an infinite number of ways, and the sum of the geometric series is given above.

The states $(0, d)$ and $(1, d)$ are reached by transitioning from state $(-1, d-1)$ with probabilities $\bar{p}_{\gamma,d}$ and $p_{\gamma,d}$, respectively. Thus, we have

$$\begin{aligned} p_{(0,d)} &= \bar{p}_{\gamma,d}p_{(-1,d-1)} \ , d = \{1, \ldots, F-1\} \ , \\ p_{(0,F)} &= \bar{p}_{\gamma,F}p_{(-1,F-1)} + \bar{p}_{\gamma,f}p_{(-1,F)} \ , \\ p_{(1,d)} &= p_{\gamma,d}p_{(-1,d-1)} \ , d = \{1, \ldots, F-1\} \ , \\ p_{(1,F)} &= p_{\gamma,F}p_{(-1,F-1)} + p_{\gamma,f}p_{(-1,F)} \ . \end{aligned}$$

The above equations reflect that the final states $(0, F)$ and $(1, F)$ can be reached both from $(0, F-1)$ and from $(0, F)$.

The states $(2, d)$, are reached only from states $(1, d)$, and we have $p_{(2,d)} = p_\alpha p_{(1,d)}$. Then, the transmission probability of a node is denoted p_{TX} and is given by

$$p_{TX} = \sum_{d=1}^{F} p_{(2,d)} \ . \tag{9}$$

At any sampling instant, the packet must be in any of the $(-1, d)$ states. Thus, we have

$$\sum_{d=0}^{F} p_{(-1,d)} = 1 \ . \tag{10}$$

A packet in any of the $(2, d)$ states will access the channel. A busy channel results when more than one such packet accesses the channel. For a channel with M scheduled nodes, the j^{th} node can find its probability of a busy channel using

$$p^j = 1 - \prod_{i \neq j, i=1}^{M} \left(1 - p^i_{TX}\right) , \tag{11}$$

where p^i_{TX} is the transmission probability of any of the other $M - 1$ nodes.

For a network with M nodes, we can write $2M$ equations (10,11) in $2M$ variables ($p_{(-1,0)}$ and p for each node). These can be solved to find the corresponding steady state solution for each node in the network. Then, we note that a node, which is successful in transmission, transitions to the state $(-1, 0)$. Thus, the probability of a successful transmission is given by $p_{(-1,0)}$ in (7). ∎

This model helps us characterize the net performance we obtain from the state-based scheduler and the CRM. Note that we use the conditional independence assumption stated earlier in (11), which simplifies the analysis.

Length of the Markov chain: The restriction on the memory of the Markov chain can be motivated by the maximum burst length of the network. Notice that the sequence $p_{(-1,d)}$, i.e. probabilities of the states $(-1, d)$ in (8), is decreasing for $p_{\gamma,d} > 0$. For an uncongested network (with a sufficiently low p), the sequence decreases rapidly, and the infinite Markov chain can be approximated by a finite Markov chain.

Computing scheduler probabilities: The scheduler probabilities $p_{\gamma,d}$ are not easy to compute, as they depend on the previous scheduler outcomes $\{\boldsymbol{\gamma}\}_{k-d+1}^{k-1}$. This is because γ cannot be inferred from the transmission history $\{\boldsymbol{\delta}\}_{k-d+1}^{k-1}$, or the delay d_k, when there is other traffic in the network. Hence, the scheduler probabilities must be computed as $p_{\gamma,d} = \sum_{\{\boldsymbol{\gamma}\}_{k-d+1}^{k-1}} \Pr(\gamma_k = 1, \{\boldsymbol{\gamma}\}_{k-d+1}^{k-1} | d_{k-1})$. This computation requires probabilities from the above Markov chain, and it can be solved along with the expressions in (8)–(11).

C. Extensions to General Models

1) CSMA with retransmissions: A more realistic MAC is likely to use retransmissions, or any other method of spreading congested network traffic over longer periods. A description of such a MAC requires a Markov chain of its own. Thus, the states $(2, d)$ in Fig. 4 can be generalized to the states $(2, d, r)$ and $(3, d, r)$ in Fig. 5, for $r \in \{1, \ldots, R\}$. Now, there are two time scales in use; the scheduler operates at the sampling period, and the CRM operates at the slot period of the MAC. The respective Markov chains can be embedded, so long as the time scales do not overlap. Thus, the CRM must complete all its R retransmissions before the next sampling instant.

However, note that for synchronized traffic with retransmissions in the CRM, the uniform independence property is not likely to hold anymore. A node in the ATT state sees the network in a steady state that is unique to each retransmission attempt. An independent CRM analysis can be performed, with an equation such as (11) for every retransmission, which

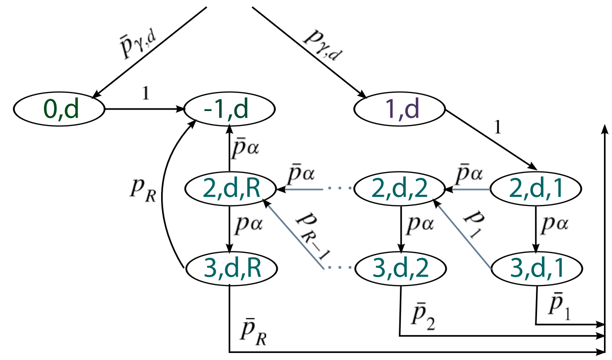

Fig. 5. Embedding a CRM with R distinct re-transmission stages in the Markov chain model

gives us

$$p^j_r = 1 - \prod_{i \neq j, i=1}^{M} \left(1 - p^i_{TX,r}\right) , \text{for } r \in \{1, \ldots, R\}$$

where, $p^i_{TX,r} = \sum_{d=1}^{F} p_{(3,d,r)}$, denotes the probability that a node is in the r^{th} retransmission attempt irrespective of the scheduler stage. However, analyzing a more complicated CRM such as CSMA/CA remains difficult, as the analysis of the protocol requires a uniform traffic distribution.

2) Asynchronous Networks: Consider an asynchronous network, with the CRM operated in a beacon-enabled mode. Now, the MAC slots are synchronized over the network, but each node can choose to sample in any MAC slot. An illustration of the behaviour, with and without retransmissions in the CRM, for synchronous and asynchronous networks, is provided in Fig. 6. For an asynchronous network with no retransmissions in the CRM, the number of interfering transmissions in the ATT state in (11) is given by $M^{(j)} < M$, where $M^{(j)}$ is the number of nodes whose sampling instants lie in the same MAC slot of the j^{th} node. Thus, the performance of the network depends on the sampling points chosen by the nodes. The more spread apart they are, the better the performance.

For an asynchronous network with retransmissions in the CRM, there is no uniform probability p. The steady state seen by a retransmission stage can only be determined by an exhaustive combinatorial analysis of the different steady

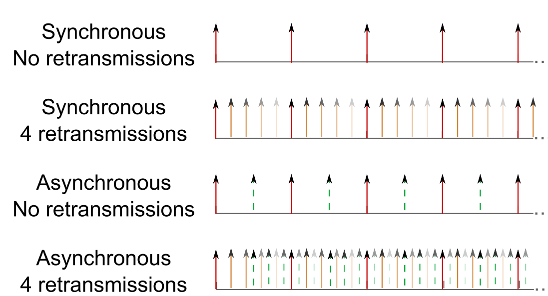

Fig. 6. A comparison of synchronous and asynchronous traffic, with and without retransmissions in the CRM. The steady state analysis differs for each traffic pattern, as nodes in the ATT states see non-uniform steady states for the last three patterns.

TABLE I

A COMPARISON OF ANALYTICAL AND SIMULATED VALUES OF p

Parameter	Simulation	Analysis
$\Pr(\delta_k = 1)$	0.1840	0.1872
p_1	0.5937	0.5944
p_2	0.5655	0.5620
p_3	0.5367	0.5277
p_4	0.5076	0.4917
p_5	0.4778	0.4542

states corresponding to the relevant retransmission stages of interfering nodes in the network.

V. SIMULATION EXAMPLE

We have run Monte-Carlo simulations for the different network configurations described above, such as CSMA with and without retransmissions, and asynchronous networks with no retransmissions, and find that the analytical value of the steady state solution matches the simulated value even in unsaturated traffic conditions, with very few retransmission stages. We present the results of one experiment here. We consider a homogenous network of $M = 10$ nodes, with $R = 5$ retransmissions in the MAC. The dynamics of the plants are given by $x_{k+1} = x_k + u_k + w_k$, where $w_k \sim \mathcal{N}(0, 1)$. We use a scheduling criterion given by $|x_k - x_{c,k}|^2 > \epsilon$, where $\epsilon = 1$, and $x_{c,k}$ denotes the comparison value at time k. It is given by $x_{c,k} = \hat{x}_{k|\tau_{k-1}}$ when the perceived delay in transmission, $d_k = k - \tau_{k-1} < F$, where $\hat{x}_{k|\tau_{k-1}} = \mathbb{E}[x_k|\mathbb{I}^C_{\tau_{k-1}}]$ is the MMSE estimate at the controller. When the perceived delay exceeds the memory of the scheduling policy, $x_{c,k}$ is given by x_{k-F}. To realize a scheduler such as this, we require the dual predictor architecture presented in [14]. The scheduler probabilities are found to be $p_{\gamma,d} = \begin{bmatrix} 0.3171 & 0.5138 \end{bmatrix}$, and $p_\alpha = 0.2$ for all 5 retransmission stages. The simulated results are provided in Table I, and they agree closely with the analytical values computed using Theorem 4. Thus, the independence assumption holds well, and can be used to motivate the Markov modelling.

A comparison of analytical and simulated values of the probability of a successful transmission, $\Pr(\delta_k = 1)$, versus the scheduler threshold, ϵ, for this synchronized network is shown in Fig. 7. The performance obtained from the network is, in accordance with expectations, poor due to synchronization and congestion. Low thresholds cause many packets to flood the network, and result in a low probability of a successful transmission due to congestion. High thresholds reduce the utilization of the network, and the probability of a successful transmission decreases again. Thus, there is a threshold that optimizes use of the network resources, and a performance analysis is required to characterize this threshold.

VI. CONCLUSIONS AND FUTURE WORKS

We have presented a method to analyze the performance of a network of event-based systems that use a CRM to access the shared network, for event-triggering policies that permit a Markovian description. We show that this Markov modelling is possible for certain event-based systems, and derive an exact joint analysis for the event-triggering policy and CRM.

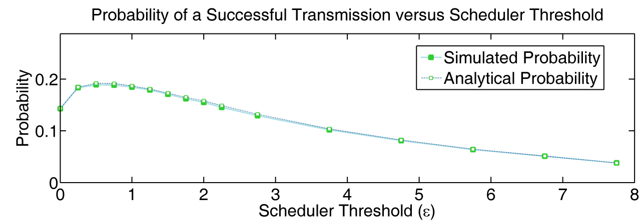

Fig. 7. A comparison of the analytical and simulated values of the probability of a successful transmission versus the scheduler threshold. Low thresholds result in a low $\Pr(\delta_k = 1)$ due to congestion. High thresholds also result in a low $\Pr(\delta_k = 1)$, but due to under-utilization of the network.

This analysis assumes conditional independence from other traffic only when the node attempts to transmit. We are able to verify this assumption through simulations, and provide extensions to more complicated network configurations. For future work, we wish to use the insights obtained through this work to simplify the performance analysis, and extend it to the general class of event-triggering policies.

VII. ACKNOWLEDGMENTS

The authors are grateful for the discussions with Pangun Park and Piergiuseppe Di Marco.

REFERENCES

[1] K. J. Åström and T. Hägglund, *PID Controllers: Theory, Design, and Tuning*. ISA, 1995.

[2] J. K. Yook, D. M. Tilbury, and N. R. Soparkar, "Trading computation for bandwidth: reducing communication in distributed control systems using state estimators," *IEEE Transactions on Control Systems Technology*, vol. 10, pp. 503 –518, July 2002.

[3] P. G. Otanez, J. R. Moyne, and D. M. Tilbury, "Using deadbands to reduce communication in networked control systems," in *Proceedings of the American Control Conference*, vol. 4, pp. 3015 – 3020 vol.4, 2002.

[4] K. J. Åström and B. M. Bernhardsson, "Comparison of Riemann and Lebesgue sampling for first order stochastic systems," in *Proceedings of the 41st IEEE Conference on Decision and Control*, vol. 2, pp. 2011 – 2016 vol.2, 2002.

[5] M. Rabi, *Packet based Inference and Control*. PhD thesis, Institute for Systems Research, University of Maryland, 2006.

[6] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1680 –1685, 2007.

[7] R. Rom and M. Sidi, *Multiple access protocols: performance and analysis*. New York, NY, USA: Springer-Verlag New York, Inc., 1990.

[8] L. Kleinrock and F. Tobagi, "Packet switching in radio channels: Part I–carrier sense multiple-access modes and their throughput-delay characteristics," *IEEE Transactions on Communications*, vol. 23, pp. 1400 – 1416, Dec. 1975.

[9] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 535 –547, Mar. 2000.

[10] A. Cervin and T. Henningsson, "Scheduling of event-triggered controllers on a shared network," in *Proceedings of the 47th IEEE Conference on Decision and Control*, pp. 3601 –3606, 2008.

[11] M. Rabi and K. H. Johansson, "Scheduling packets for event-triggered control," *Proceedings of 10th European Control Conference*, pp. 3779 – 3784, 2009.

[12] T. Henningsson and A. Cervin, "A simple model for the interference between event-based control loops using a shared medium," in *Proceedings of the 49th IEEE Conference on Decision and Control*, 2010.

[13] R. Blind and F. Allgöwer, "Analysis of networked event-based control with a shared communication medium: Part I - Pure ALOHA," *Submitted to IFAC World Congress*, 2011.

[14] C. Ramesh, H. Sandberg, and K. H. Johansson, "LQG and medium access control," *Preprints of the 1st IFAC Workshop on Estimation and Control of Networked Systems*, pp. 328 – 333, September 2009.