2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Timeout Control in Distributed Systems Using Perturbation Analysis

Ali Kebarighotbi and **Christos G. Cassandras**
Division of Systems Engineering
and Center for Information and Systems Engineering
Boston University
Brookline, MA 02446
alik@bu.edu,cgc@bu.edu

***Abstract*— Timeout control is a simple mechanism used when direct feedback is either impossible, unreliable, or too costly, as is often the case in distributed systems. Its effectiveness is determined by a timeout threshold parameter and our goal is to quantify the effect of this parameter on the behavior of such systems. In this paper, we consider a basic communication system with timeout control, model it as a stochastic flow system, and use Infinitesimal Perturbation Analysis to determine the sensitivity of a "goodput" performance metric with respect to the timeout threshold parameter. In conjunction with a gradient-based scheme, we show that we can determine an optimal value of this parameter. Some numerical examples are included.**

I. Introduction

Timeout control is a simple mechanism used in many systems where direct feedback is either impossible, unreliable, or too costly. A timeout event is scheduled using a *timer* which expires after some *timeout threshold* parameter. This defines an expected time by which some other event should occur. If no information arrives within this period, a "timeout event" occurs and incurs certain reactions which are an integral part of the controller. This simple reactive control policy has been used for stabilizing systems ranging from manufacturing to communication systems [12], Dynamic Power Management (DPM) [3],[11],[21] and software systems [8],[9] among others. Despite its wide usage, quantifications of its effect on system behavior have not yet received the attention they deserve. In fact, timeout controllers are usually designed based on heuristics which may lead to poor results; e.g., see [1],[12],[20]. There is limited work intended to find optimal timeout thresholds; for example, in Automatic Guided Vehicles [19], DPM [11], [3], [4], and finally communication systems [10],[16],[17]. All such approaches are limited by their reliance on the distributional information about the stochastic processes involved. In distributed systems, where usually control decisions must be made with limited information from remote components, timeouts provide a key mechanism through which a controller can infer valuable information about the unobservable system states. In fact, as pointed out in [28], timeouts are indispensable tools in building up reliable distributed systems. Defining θ_i as the timeout threshold associated with an action A_i, a proper response to a timeout event is normally related to the state and action at time $t - \theta_i$. A simple example is repeating A_i at t because no desirable response has been observed in $[t - \theta_i, t)$. Thus, a controller must have some information about both current and past states and actions of the system. Figure 1 shows a high-level model for timeout-controlled distributed systems. The control is shown as the input $u(S, z)$ where S is the information available to the controller, consisting of its own state X_c, observable system state X_1, and their associated histories $\tilde{X}_c$ and $\tilde{X}_1$. The feedback signal z carries information about the state of the remote system and is subject to random communication delays. The controller adjusts $u(S, z)$ as a result of its input: z and timeout events that it generates. We view the overall controller-system model as a general *Stochastic Hybrid System* (SHS), where the "system" may be time-driven, event-driven or hybrid in itself. Note that the "controller" in Fig. 1 may be located at a central point communicating with remote components or locally at each component with its own view of the rest of the "system."

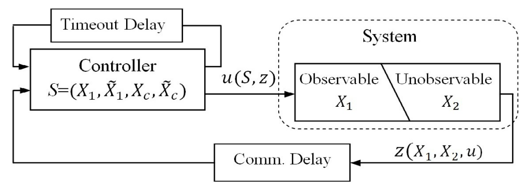

Fig. 1. Timeout controlled distributed system

In this paper, we set forth a line of research aimed at quantifying how timeout threshold parameters affect the system state and ultimately its behavior and performance. Here, we limit ourselves to a setting in Fig. 1 where the "system" is a Discrete Event System (DES), so that the controller output includes a response to either a timeout event or an event carrying information through z. However, since stochastic DES models can be very complicated to analyze, we rely on recent advances which abstract a DES into a SHS and, in particular, the class of *Stochastic Flow Models* (SFMs). A SFM treats the event rates as stochastic processes of arbitrary generality except for mild technical conditions. Many performance gradient estimates can be obtained through IPA techniques for general SHS [7],[24],[22],[27],[13],[15],[6],[23]. In addition, a fundamen-

This work is supported in part by NSF under Grant EFRI-0735974, by AFOSR under grant FA9550-09-1-0095, by DOE under grant DE-FG52-06NA27490, and by ONR under grant N00014-09-1-1051.

tal property of IPA in SFMs (as in DES) is that the derivative estimates obtained are independent of the probability laws of the stochastic rate processes and require minimal information from the observed sample path. Our goal here is to use a similar approach for online optimal timeout control in various systems within the framework of Fig. 1. As a starting point, we adopt a basic communication system to show how a SFM of a timeout-controlled distributed system can be obtained and then used to optimize the timeout threshold.

The contributions of this work are twofold: (a) We believe this is the first effort to combine a SFM with IPA to find optimal timeout thresholds. Two related attempts in [25],[18] used buffer capacity to control the average timeout rate; here, we directly control the timeout parameter and handle the fact that the associated SFM involves stochastic delay differential equations. (b) We extend previous works on timeout control by removing any dependence on distributional information.

This paper is organized as follows. In Section II we present the DES we consider and obtain a stochastic timed automaton model. Section III is dedicated to the stochastic flow abstraction of this DES, where the associated SFM is obtained. Our IPA results are included in Section IV, leading to a determination of an optimal timeout threshold for a "goodput" metric. Numerical examples are shown in Section V and we conclude with Section VI.

II. Timeout Control in a Communication Link

Consider a network link consisting of a transmitter and a receiver node connected through a channel with random transmission delays. After each packet is sent to the receiver, the transmitter expects an *acknowledgement* (shortly, ACK) from the receiver to indicate its receipt. It keeps a copy of each sent packet and initiates a *Retransmission TimeOut* (RTO) timer expecting an ACK while this timer is running down. The RTO timer is assumed to start with an initial value θ known as the *timeout threshold*. While the transmitter receives the ACKs in a timely fashion, it keeps on transmitting more packets according to some transmission policy π_1. However, when a RTO timer goes off while its associated ACK is not received by the transmitter, a *timeout event* occurs which is a strong indication of network overload and congestion. In this case, the transmitter switches to a back-off transmission policy π_2 in order to reduce the network load. This is usually done by employing a mechanism to reduce the transmission rate helping the network to alleviate congestion. The system switches back to π_1 as soon as the transmitter receives a timely ACK. This is a scheme used in almost all communication protocols, e.g. in TCP [1]. We make the following two assumptions:

Assumption 1. The communication link is lossless.

Assumption 2. There are always new data to be transmitted.

Assumptions 1,2 should not be seen as limitations of the analysis, since we can remove them at the expense of more state variables complicating the exposition. By Assumption 1, a timeout event doubly degrades channel performance metrics like "goodput" (the throughput fraction that excludes useless traffic): first, with each timeout event, a previously sent packet should be retransmitted limiting the rate of new data transmissions; second, since the timed-out packets are still in the network and use resources, they slow down other packets in the channel. A queueing model

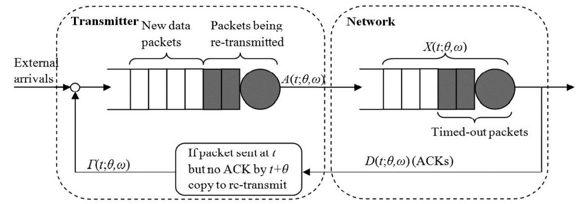

Fig. 2. System model with timeout control. Timed-out packets and corresponding packets to re-transmit are shown in grey.

of the system described above is shown in Fig. 2. The transmitter has a priority queue processing re-transmitted packets first, while the network queue has a simple FIFO mechanism. Considering a sample path of this DES over a finite interval $[0, T]$ indexed by ω, at each time $t \in [0, T]$ let the number of transmitted packets be $A(t; \theta, \omega)$ and the number of departures (ACKs) the transmitter has observed be $D(t; \theta, \omega)$. Hence, the number of transmitted packets sent but not yet acknowledged at each time is defined as

$$X(t; \theta, \omega) = A(t; \theta, \omega) - D(t; \theta, \omega), \quad t \in [0, T]. \tag{1}$$

where $X(t; \theta, \omega) \in \{0, 1, \ldots\}$. We call a time period in which $X(t; \theta, \omega) > 0$, a *Non-Empty Period* (NEP); otherwise, we call it an *Empty Period* (EP) associated with the network queue in Fig. 2. We model the transmitter with a never-empty buffer whose service (or transmission) process is governed by the policies π_1 and π_2. The round-trip channel delay (the time from the transmission of a packet until the receipt of its associated ACK) is modeled using a queue with arbitrary service distribution. We have modeled the timeout mechanism by making a copy from the timed-out packet and putting it back in the transmitter buffer with a higher priority. We denote the number of timeout events that have occurred in $[0, t) \subset [0, T]$ by $\Gamma(t; \theta, \omega)$.

The purpose of this paper is to find an optimal timeout threshold θ^* maximizing the communication link *goodput* at the transmitter, defined as

$$J_T(\theta) = \mathrm{E}_\omega[L(T, \theta, \omega)] = \mathrm{E}_\omega[A(T, \theta) - 2\Gamma(T, \theta)], \tag{2}$$

where $\mathrm{E}_\omega[\cdot]$ is the expectation operator. Below, we write $L(\theta)$ and omit T, ω for simplicity. We also omit the sample path index ω from the arguments of all stochastic processes.

Associated with the DES of Fig. 2 is an *event set* $\mathcal{E} = \{a_1, a_2, q_1, q_2, d\}$ where a_1, a_2 are transmission events while the system operates according to policies π_1, π_2, respectively. Regardless of policy, with each transmission at time t a *potential timeout* is scheduled at $t + \theta$. If the ACK for this transmission is observed within $[t, t + \theta)$, the potential timeout scheduled becomes a *disabled timeout* event, denoted by q_1; otherwise, it is an *actual timeout*, denoted by q_2. Both q_1 and q_2 help us determine which policy should be used. Finally, d is the event of an ACK receipt by the transmitter.

To determine which of the the policies π_1, π_2 should be used at any time, we introduce a new state variable

$$Y(t;\theta) = A(t-\theta;\theta) - D(t;\theta), \quad t \in [0,T]. \tag{3}$$

where $Y(t;\theta) \in \{\ldots,-1,0,1,\ldots\}$ and we assume the information on $A(t;\theta)$ for $t \in [-\theta, 0)$ is available at $t=0$. Notice that since $A(t;\theta)$ is a non-decreasing function of t, we have $X(t;\theta) - Y(t;\theta) = A(t;\theta) - A(t-\theta;\theta) \geq 0$ for all $t \in [0,T]$. If $Y(t;\theta) > 0$, some of the packets sent by time $t-\theta$ are still in the network queue at t and unacknowledged, hence $Y(t;\theta)$ is the *current number of timed-out packets* in the network. Otherwise, $|Y(t;\theta)|$ is the *number of disabled potential timeouts* in $[t-\theta, t)$. Accordingly, we say the system is in *timeout mode* whenever $Y(t;\theta) > 0$ and in *normal mode*, otherwise. We also partition the sample path into *Normal Periods* (NPs) and *TimeOut Periods* (TOPs).

III. Stochastic Flow Abstraction

We use the systematic road-map proposed in [26], giving explicit criteria for each step of abstraction to ensure consistency between the behavior of the resulting SHS and the original DES. To this end, we first define a *Stochastic Timed Automaton* (STA) [5] for the DES, i.e., a quintuple $(\mathcal{S}, \mathcal{E}, \phi, \Psi, G)$ where $\mathcal{S}$ is a countable discrete state space, $\mathcal{E}$ is the event set, $\phi : \mathcal{S} \times \mathcal{E} \mapsto \mathcal{S}$ is a transition function, and $\Psi : \mathcal{S} \mapsto 2^{\mathcal{E}}$ is the *active event set* function defining the feasible events at each discrete state $S \in \mathcal{S}$. Finally, G is a *clock structure* containing the cumulative distribution functions of all event lifetimes for each $e \in \mathcal{E}$.

Given a STA, a *partition operator* is a mapping $P : \mathcal{S} \mapsto \mathbb{Z}$ dividing the state space $\mathcal{S}$ into non-overlapping and non-empty subsets. Accordingly, two states S_1 and S_2 are in the same subset iff $P(S_1) = P(S_2)$. Moreover, S is called an *interior state* of a subset $\mathrm{P} \subset \mathcal{S}$ (denoted by $S \in \mathrm{P}^o$) if for all feasible events $e \in \Psi(S)$, we have $P(\phi(S,e)) = P(S) = \mathrm{P}$; otherwise, it is a *boundary state*. Accordingly, a *boundary mode* m is one where $\mathrm{P}^o_m = \emptyset$. Following [26], there are two criteria that guide the construction of a *Hybrid Automaton* from a given Timed Automaton:

- C1. For any two states $S_1, S_2 \in \mathcal{S}$, if $P(S_1) = P(S_2) = \mathrm{P}$, then $\Psi(S_1) = \Psi(S_2) = \Psi(P)$.
- C2. For each $e \in \Psi(\mathrm{P})$, the state transition function $\phi(S,e)$ must be non-piecewise [2] in all $S \in \mathrm{P}^o$.

Criterion C1 requires that all states in a subset P share the same feasible event set. Criterion C2 implies that every feasible event in an interior state $S \in \mathrm{P}^o$ must have the same effect on all the states within the subset's interior P^o. For example, $\phi(S,e) = S+1$ for every $S \in \mathrm{P}^o$ and some $e \in \Psi(S) = \Psi(\mathrm{P})$ satisfies this condition. A desired subset is the largest set satisfying criteria C1 and C2. Considering our case, we have already defined $\mathcal{E} = \{a_1, a_2, q_1, q_2, d\}$ and $\mathcal{S}$ is defined by the state pair $S(t) = (X(t), Y(t))$. Thus, using (1) and (3) to identify the effect of each event $e \in \mathcal{E}$ on each state S, the STA of the DES is shown in Fig. 3. Next, applying the partitioning criteria C1 and C2 to this STA, we identify five aggregate sets (modes) as follows:

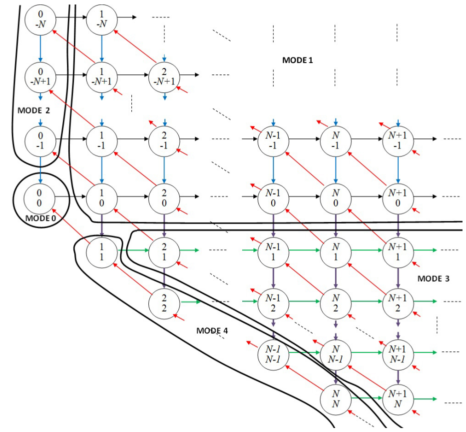

Fig. 3. Stochastic Timed Automaton (STA) model of the DES. Transition events: a_1-Black; a_2-Green; q_1-Blue; q_2-Purple; d-Red.

$$M(t;\theta) = \begin{cases} 0 & \text{if } X(t;\theta) = Y(t;\theta) = 0 \\ 1 & \text{if } X(t;\theta) > 0,\ Y(t;\theta) \leq 0 \\ 2 & \text{if } X(t;\theta) = 0,\ Y(t;\theta) < 0 \\ 3 & \text{if } X(t;\theta) > Y(t;\theta) > 0 \\ 4 & \text{if } X(t;\theta) = Y(t;\theta) > 0 \end{cases} \tag{4}$$

with $\Psi(\mathrm{P}_0) = \{a_1\}$, $\Psi(\mathrm{P}_1) = \{a_1, q_1, d\}$, $\Psi(\mathrm{P}_2) = \{a_1, q_1\}$, $\Psi(\mathrm{P}_3) = \{a_2, q_2, d\}$ and $\Psi(\mathrm{P}_4) = \{a_2, d\}$. Abstracting the DES into a SFM consists of five steps:

1) Abstracting DES states: We assign the continuous state flow processes $\{x(t;\theta)\}$ and $\{y(t;\theta)\}$ to the discrete states $X(t;\theta)$ and $Y(t;\theta)$, respectively. Also, we replace $X(t;\theta), Y(t;\theta)$ by $x(t;\theta), y(t;\theta)$ in (4).

2) Abstracting events into flow rates: For all $t \in [0,T]$ we define the *transmission flow rate* processes $\{\alpha_i(t;\theta)\}, i = 1,2$ as the flows associated with the a_1 and a_2 events, respectively. When the emphasis is only on time rather than the transmission policy, we will write $\alpha(t;\theta)$ with the understanding that $\alpha(t;\theta) = \alpha_i(t;\theta)$ if policy π_i applies as $t \in [0,T]$. Next, we define the *actual network discharge process* $\{d(t;\theta)\}$ associated with the event d in the DES by defining the *maximal network discharge rate* process $\{\beta(t)\}$ which is independent of θ. Then, considering Fig. 2, when $X(t;\theta) > 0$ the departure process is dictated by the network processing rate, i.e., $d(t;\theta) = \beta(t)$ when $x(t;\theta) > 0$; when $x(t;\theta) = 0$, we have $d(t;\theta) = \alpha(t;\theta) \leq \beta(t)$. Finally, we let the *disabled timeout flow rate* process $\{\gamma_1(t;\theta)\}$ and *actual timeout flow rate* process $\{\gamma_2(t;\theta)\}$ abstract the occurrence frequencies of the events q_1 and q_2, respectively. Since, in the DES, q_1 and q_2 events occur after a time delay θ from their associated transmission events, we set:

$$\gamma_1(t;\theta) = \begin{cases} \alpha(t-\theta;\theta) & \text{if } M(t) = 0,1,2 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

$$\gamma_2(t;\theta) = \begin{cases} \alpha(t-\theta;\theta) & \text{if } M(t) = 3,4 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

3) Obtaining (differential) flow equations for SFM modes: Using the state equations (1) and (3) it is easy to derive the associated flow conservation equations that dictate the time-driven dynamics in each SFM mode. Omitting details (however, see [14]) we get

$$\dot{x}(t;\theta) = f_x(t;\theta) = \begin{cases} 0 & M(t) = 0,2 \\ \alpha(t;\theta) - \beta(t) & \text{otherwise} \end{cases} \tag{7}$$

$$\dot{y}(t;\theta) = f_y(t;\theta) = \begin{cases} 0 & M(t) = 0 \\ \alpha(t-\theta;\theta) - \beta(t) & M(t) = 1,3 \\ \alpha(t-\theta;\theta) - \alpha(t;\theta) & M(t) = 2 \\ \alpha(t;\theta) - \beta(t) & \text{otherwise} \end{cases} \tag{8}$$

4) Obtaining mode transitions: Considering Fig. 3, along with (4), (7), and (8), we can readily determine the *exogenous* events or *endogenous* events (guard conditions) that define all mode transitions. Recall (see [7]) that an event occurring at time τ_k is (i) *Exogenous* if it causes a discrete state transition independent of $\theta \in \Theta$ and satisfies $\frac{d\tau_k}{d\theta} = 0$, and (ii) *Endogenous*, if there exists a continuously differentiable function $g_k : \mathbb{R}^n \times \Theta \to \mathbb{R}$ such that $\tau_k = \min\{t > \tau_{k-1} : g_k(x(\theta,t),\theta) = 0\}$. We define an event set of the SFM as

$$\mathcal{E}_{SFM} = \{e_\beta, [x(t;\theta) = 0], [y(t;\theta) = 0], [x(t;\theta) = y(t;\theta)], [\alpha(t;\theta) > \alpha(t-\theta;\theta)], [\alpha(t;\theta) > \beta(t)]\}$$

where e_β is an *exogenous* event occurring if there is an uncontrollable jump in $\beta(t)$, and the remaining are all *endogenous* events each corresponding to a condition associated with some $g_k(x(\theta,t),\theta) = 0$ as defined above. The last event can be due to e_β but this case is already covered. Thus, we consider $[\alpha(t;\theta) > \beta(t)]$ as endogenous.

From (7) and Fig. 3 the transitions $0 \to 1$ and $2 \to 1$ occur when $\alpha(t;\theta) - \beta(t)$ switches from ≤ 0 to > 0, i.e., when $[\alpha(t;\theta) > \beta(t)]$ occurs or e_β causes this sign change. Conversely, returning to $M(t) = 0,2$ in (7) involves $4 \to 0$ and $1 \to 2$ which are due to $[x(t;\theta) = 0]$. Other cases can be similarly analyzed (See [14] for details). Figure 4 shows the Stochastic Hybrid Automaton (SHA) of the SFM.

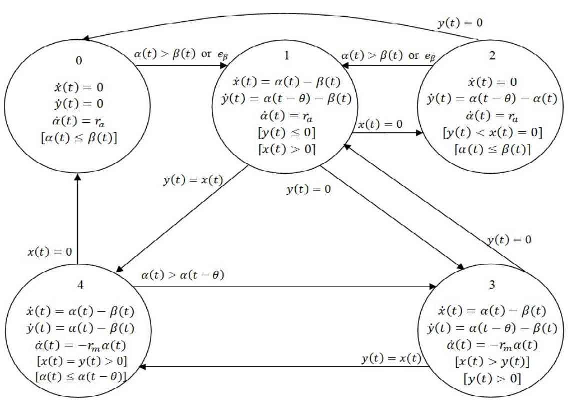

Fig. 4. Stochastic hybrid automaton for the SFM.

5) Abstracting the clock structure: Our analysis is independent of policies π_1 or π_2 but in order to get concrete results and motivated by the common Additive Increase Multiplicative Decrease (AIMD) schemes, we let

$$\dot{\alpha}(t;\theta) = f_\alpha(t;\theta) = \begin{cases} r_a & \text{if } M(t) = 0,1,2 \\ -r_m\alpha(t;\theta) & \text{otherwise} \end{cases} \tag{9}$$

where $r_a > 0$, $r_m > 0$ are chosen such that $\alpha(t;\theta)$ matches the transmission rate in the DES as close as possible. We assume the initial condition $\alpha(0;\theta) = 0$ for simplicity.

Lemma 1. If π_1 and π_2 are given by (9), the transition $4 \to 3$ is infeasible.

All proofs are omitted but can be found in [14].

Remark 1: This result is a byproduct of the choice of policies in (9) and not the underlying SFM. That is, the transition $4 \to 3$ is in general feasible for other π_1, π_2.

Finally, returning to the goodput performance metric in (2), since $\gamma_2(t;\theta)$ is the rate at which the flow content in the network times out, we write the SFM version of (2) as

$$J(\theta) = \mathrm{E}\left[\int_0^T [\alpha(t;\theta) - 2\gamma_2(t;\theta)]dt\right] \tag{10}$$

IV. Infinitesimal Perturbation Analysis

Let $\{\tau_k(\theta)\}$, $k = 1,\ldots,K$, denote the event times in a state trajectory. For convenience, we set $\tau_0 = 0$ and $\tau_{k+1} = T$. Next, let $\tau_k' = \frac{\partial \tau_k}{\partial \theta}$. More generally, we use $z' \equiv \frac{\partial z}{\partial \theta}$ for all variables $z(\theta)$ that depend of θ. To keep the notation manageable, we drop θ from function arguments except for $J(\theta)$ and $L(\theta,\omega)$. The purpose of IPA is to estimate $\frac{dJ(\theta)}{d\theta} = \frac{d\mathrm{E}[L(\theta,\omega)]}{d\theta}$ by means of $\frac{dL(\theta,\omega)}{d\theta}$ which is an unbiased estimate of $\frac{dJ(\theta)}{d\theta}$ under certain (generally mild) conditions [7].

Recall, from (6), that $\gamma_2(t) = 0$ except when $M(t) = 3,4$. Let $\mathcal{T} \subset [0,T]$ such that $t \in \mathcal{T}$ iff $M(t) \in \{3,4\}$ and set $\mathcal{T} = \cup_{i=1}^N \mathcal{T}_i$ for some $N < K$, i.e., $\mathcal{T}_i$ is the ith TOP in a sample path over $[0,T]$. For each $\mathcal{T}_i$, define the index set $\Lambda_i = \{k : [\tau_k, \tau_{k+1}) \in \mathcal{T}_i\}$. We can then write:

$$L(\theta,\omega) = \sum_{k=0}^{K} \int_{\tau_k}^{\tau_{k+1}} \alpha(t)dt - 2\sum_{i=1}^{N} \sum_{k\in\Lambda_i} \int_{\tau_k}^{\tau_{k+1}} \alpha(t-\theta)dt.$$

Omitting details (however, see [14] for complete analysis), we can write the IPA derivative as

$$\frac{dL(\theta,\omega)}{d\theta} = \sum_{k=0}^{K} \alpha'(t)dt - 2\sum_{i=1}^{N} \sum_{k\in\Lambda_i} \int_{\tau_k}^{\tau_{k+1}} \alpha'(t-\theta)dt + 2\sum_{i=1}^{N} \left[(1-\tau_{u_i}')\alpha(\tau_{u_i}-\theta) - (1-\tau_{l_i}')\alpha(\tau_{l_i}-\theta)\right] \tag{11}$$

where l_i and u_i are the first and last indices in Λ_i.

Before proceeding, we provide a brief review of the IPA framework for general stochastic hybrid systems as presented in [7]. If $s(t)$ is the state vector of the SFM, IPA specifies how changes in θ influence $s(t)$ and the event times τ_k and, ultimately, how they influence interesting performance metrics. Let us assume that over an interval $[\tau_k, \tau_{k+1})$, the SFM is at some mode during which the time-driven state satisfies $\dot{s} = f_k(s,\theta,t)$ for some $f_k : \mathbb{R}^n \times \mathbb{R}^m \times [0,T) \to \mathbb{R}^n$. Let $s'(t) \equiv \frac{\partial s(t)}{\partial \theta} \in \mathbb{R}^n \times \mathbb{R}^m$ be the Jacobian matrix for all state derivatives. It is shown in [7] that, for any $t \in [\tau_k, \tau_{k+1})$, $s'(t)$ satisfies:

$$\frac{d}{dt}s'(t) = \frac{\partial f_k(t)}{\partial s}s'(t) + \frac{\partial f_k(t)}{\partial \theta} \tag{12}$$

for $t \in [\tau_k, \tau_{k+1})$ with boundary condition:

$$s'(\tau_k^+) = s'(\tau_k^-) + \tau_k'\left[f_{k-1}(\tau_k^-) - f_k(\tau_k^+)\right] \tag{13}$$

for $k = 0, \ldots, K$. For an exogenous event e_k at τ_k, $\tau_k' = 0$ but for every endogenous event we have a continuously differentiable function $g_k : \mathbb{R}^n \times \Theta \to \mathbb{R}$ such that $\tau_k = \min\{t > t_{k-1} \; : \; g_k\left(s(t;\theta), \theta\right) = 0\}$. It is shown in [7] that

$$\tau_k' = -\left[\frac{\partial g_k}{\partial s} f_k(\tau_k^-)\right]^{-1}\left(\frac{\partial g_k}{\partial \theta} + \frac{\partial g_k}{\partial s} s'(\tau_k^-)\right) \tag{14}$$

if $e_k \in \mathcal{E}_{SFM}$ at τ_k is endogenous and defined as long as $\frac{\partial g_k}{\partial s} f_k(\tau_k^-) \neq 0$. Let us define $s(t) = [\alpha(t), \alpha(t-\theta), \beta(t), x(t), y(t)]^{\mathrm{T}}$ so that for a switching function $g_k\left(s(t;\theta), \theta\right) = 0$ we have $\frac{\partial g_k}{\partial s} = [\frac{\partial g_k}{\partial \alpha(t)}, \frac{\partial g_k}{\partial \alpha(t-\theta)}, \ldots, \frac{\partial g_k}{\partial y(t)}]$.

A. Event-time Derivatives: We find τ_k' for each event in $\mathcal{E}_{SFM}$. First, $\tau_k' = 0$ for the exogenous event $e_\beta \in \mathcal{E}_{SFM}$. The following lemma derives τ_k' for the endogenous events:

Lemma 2. Under policies π_1, π_2 defined in (9), if e_k is

$$(i)[\alpha(\tau_k) > \beta(\tau_k)]: \quad \tau_k' = \frac{-\alpha'(\tau_k^-)}{r_a - \dot{\beta}(\tau_k^-)}$$

$$(ii)[x(\tau_k) = 0]: \quad \tau_k' = \frac{-x'(\tau_k^-)}{\alpha(\tau_k) - \beta(\tau_k)}$$

$$(iii)[y(\tau_k) = 0]: \; \tau_k' = \begin{cases} \frac{-y'(\tau_k^-)}{\alpha(\tau_k-\theta)-\alpha(\tau_k)} & \text{if } \phi(2, e_k) = 0 \\ \frac{-y'(\tau_k^-)}{\alpha(\tau_k-\theta)-\beta(\tau_k)} & \text{otherwise} \end{cases}$$

$$(iv)[x(\tau_k) = y(\tau_k)]: \quad \tau_k' = \frac{x'(\tau_k^-) - y'(\tau_k^-)}{\alpha(\tau_k - \theta) - \alpha(\tau_k)}$$

B. State Derivatives: In view of (11) and Lemma 2, we determine the state derivatives $\alpha'(t)$, $\alpha'(t-\theta)$, $x'(t)$ and $y'(t)$, $t \in [0, T]$. Since $\alpha(t-\theta)$ is only a shifted version of $\alpha(t)$, it boils down to finding $\alpha'(t)$, $x'(t)$, and $y'(t)$ and saving $\alpha'(t)$ until $t+\theta$ to be used for $\alpha'(t-\theta)$. This can be done by solving (12) for $t \in [\tau_k, \tau_{k+1})$, $k = 0, \ldots, K-1$, with initial conditions provided by (13). Starting with $\alpha(t)$,

$$\alpha(t) = \alpha(\tau_k) + \int_{\tau_k}^{t} f_\alpha(\tau) d\tau$$

which after differentiating with respect to θ gives

$$\alpha'(t) = \alpha'(\tau_k^+) + \int_{\tau_k}^{t} f_\alpha'(\tau) d\tau. \tag{15}$$

By (9), $f_\alpha'(\tau)$ is

$$f_\alpha'(\tau) = \begin{cases} 0 & \text{if } M(\tau) = 0, 1, 2 \\ -r_m \alpha'(\tau) & \text{otherwise} \end{cases} \tag{16}$$

Rewriting $f_\alpha'(\tau) = \frac{\partial}{\partial \theta}\frac{d\alpha(t)}{dt} = \frac{d}{dt}\alpha'(\tau)$, (16) shows that in NPs, $\alpha'(t)$ remains constant, whereas in TOPs, it decays exponentially with factor r_m. Similarly, for any $t \in [\tau_k, \tau_{k+1})$ and any $\tau \in [\tau_k, t)$ we find

$$f_x'(\tau) = \begin{cases} 0 & \text{if } M(\tau) = 0, 2 \\ \alpha'(\tau) & \text{otherwise} \end{cases}$$

$$f_y'(\tau) = \begin{cases} 0 & \text{if } M(\tau) = 0 \\ \alpha'(\tau - \theta) - \dot{\alpha}(\tau - \theta) & \text{if } M(\tau) = 1, 3 \\ \alpha'(\tau - \theta) - \dot{\alpha}(\tau - \theta) - \alpha'(\tau) & \text{if } M(\tau) = 2 \\ \alpha'(\tau) & \text{otherwise} \end{cases}$$

Next, we use (13) which provides the initial conditions for the state derivatives at the beginning of each interval $[\tau_k, \tau_{k+1})$. First, if τ_k is exogenous we get $\tau_k' = 0$ which eliminates the jump terms in (13). Thus, we only concentrate on the endogenous events. Below, we make use of the definition $r(\tau_k) = r_a + r_m \alpha(\tau_k)$. We also use the transition function $\phi(m_1, e) = m_2$, $e \in \mathcal{E}_{SFM}$ to indicate that while in mode m_1, e causes a transition into mode m_2.

Lemma 3. For any event time τ_k,

$$\alpha'(\tau_k^+) = \alpha'(\tau_k^-) + \begin{cases} \frac{-y'(\tau_k^-)}{\alpha(\tau_k-\theta)-\beta(\tau_k)} r(\tau_k) & \text{if } \phi(1, e_k) = 3 \\ \frac{y'(\tau_k^-)}{\alpha(\tau_k-\theta)-\beta(\tau_k)} r(\tau_k) & \text{if } \phi(3, e_k) = 1 \\ \frac{x'(\tau_k^-)-y'(\tau_k^-)}{\alpha(\tau_k-\theta)-\alpha(\tau_k)} r(\tau_k) & \text{if } \phi(1, e_k) = 4 \\ \frac{x'(\tau_k^-)}{\alpha(\tau_k)-\beta(\tau_k)} r(\tau_k) & \text{if } \phi(4, e_k) = 0 \\ 0 & \text{otherwise} \end{cases}$$

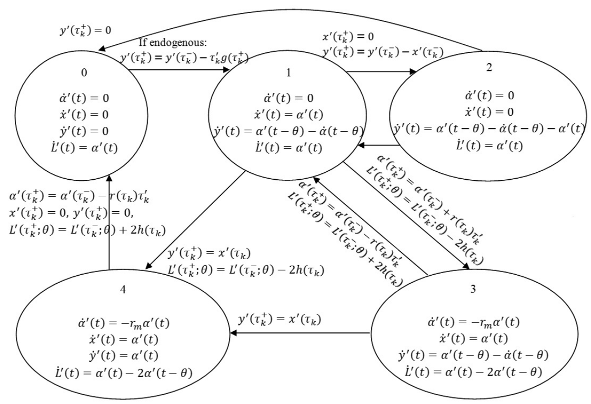

Fig. 5. Stochastic hybrid automaton for IPA estimator.

Lemma 4. For any event time τ_k,

$$x'(\tau_k^+) = \begin{cases} 0 & \text{if } \phi(4, e_k) = 0 \text{ or } \phi(1, e_k) = 2 \\ x'(\tau_k^-) & \text{otherwise} \end{cases}$$

Lemma 5. For any event time τ_k,

$$y'(\tau_k^+) = \begin{cases} y'(\tau_k^-) + \frac{\alpha'(\tau_k)[\alpha(\tau_k-\theta)-\beta(\tau_k)]}{r_a - \dot{\beta}(\tau_k)} & \text{if } \phi(0, e_k) = 1 \\ y'(\tau_k^-) - x'(\tau_k^-) & \text{if } \phi(1, e_k) = 2 \\ y'(\tau_k^-) & \text{if } \phi(1, e_k) = 3 \\ & \text{or } \phi(3, e_k) = 1 \\ x'(\tau_k^-) & \text{if } \phi(1, e_k) = 4 \\ & \text{or } \phi(3, e_k) = 4 \\ 0 & \text{otherwise} \end{cases}$$

To summarize, the IPA estimator in (11) requires the event time derivatives in Lemma 2 and $\alpha'(\tau)$, $x'(\tau)$ and $y'(\tau)$ whose initial conditions are given through Lemmas 3,4,5. Defining $h(\tau_k) = (1 - \tau_k')\alpha(\tau_k - \theta)$, a hybrid automaton is shown in Fig. 5 that captures this estimator as a SHS operating in parallel with the one in Fig. 4 and includes the IPA derivative itself shown as $\dot{L}'(t)$. Note that despite the need for information about the SFM state at $t - \theta$, implementing the IPA estimator is simple.

V. NUMERICAL EXAMPLES

We present an example where the IPA estimator is implemented in conjunction with a gradient-based optimization algorithm of the form $\theta_{n+1} = \theta_n - \eta_n H_n(\theta_n, \omega)$, where $H_n(\theta_n, \omega)$ is the IPA estimate of $dJ/d\theta$ based on a sample path ω with the timeout threshold set at θ_n and using an appropriate step size sequence $\{\eta_n\}$. This allows us to determine the optimal timeout value θ^* that maximizes the average goodput defined as $\bar{J}(\theta) = \frac{1}{T}J(\theta)$. The transmission

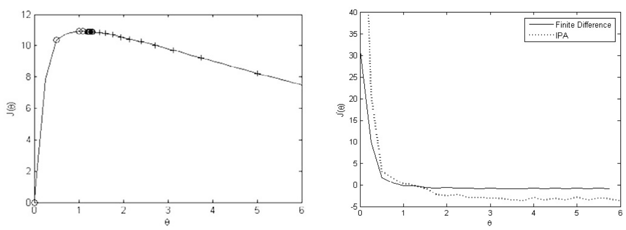

Fig. 6. Left: $\bar{J}(\theta)$ and the results of the optimization for starting points $\theta = 0$ and 5. Right: IPA vs. FD derivative

policy follows the AIMD scheme (9). In particular, in a NP, starting from 1, the transmitter increases its rate with a factor $r_a = 1$ while with each timeout it halves this rate. To approximate this in the SFM we used $r_a = 1$ and $r_m = -\log(0.5)$. The network service process is chosen as exponential with rate $\mu = 20$. For each $\theta \in \{0, 0.25, \ldots, 6\}$, we estimated the average goodput and obtained the IPA derivatives by averaging the results of 5 sample paths with length $T = 24000$ each. We applied the IPA estimator using the actual DES sample path data, as every event $e \in \mathcal{E}_{SFM}$ has a directly observable DES counterpart. The goodput performance function is shown in Fig. 6. It has a maximum at $\theta^* \approx 1$ with $J(\theta^*) = 10.88$. Figure 6 shows the IPA derivatives of the performance function along with finite difference (FD) approximations. The IPA estimator crosses $J'(\theta) = 0$ at $\theta \approx 1.25$. For both starting points, the results of our optimization algorithm show that the algorithm converges to $\theta \approx 1.25$ in under 12 iterations.

VI. CONCLUSIONS

We considered the problem of timeout control in distributed systems. Posing the problem in a general stochastic hybrid setting, we have studied a basic communication system with timeout control as a DES abstracted into a SFM. We used IPA to estimate the sensitivity of a "goodput" performance metric with respect to the timeout threshold parameter. Finally, using a gradient-based scheme, we showed that we can determine an optimal value of this parameter.

REFERENCES

[1] M. Allman, V. Paxson, and W. Stevens. TCP congestion control. *RFC2581*, April 1999.

[2] G. Arfken. *Mathematical Methods for Physicists*. Academic Press, 1985.

[3] L. Cai and Y. Hsiang Lu. Joint power management of memory and disk. In *Proc. of the Conf. on Design, Automation and Test in Europe. IEEE Computer Society*, pages 86–91, 2005.

[4] X. R. Cao and H. F. Chen. Perturbation realization, potentials, and sensitivity analysis of markov processes. *Automatic Control, IEEE Tran. on*, 42(10):1382 –1393, oct. 1997.

[5] C. G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[6] C. G. Cassandras, G. Sun, C. G. Panayiotou, and Y. Wardi. Perturbation analysis and control of two-class stochastic fluid models for communication networks. *IEEE Trans. on Automatic Control*, 48(5):770–782, 2003.

[7] C. G. Cassandras, Y. Wardi, C. G. Panayiotou, and C. Yao. Perturbation analysis and optimization of stochastic hybrid systems. *European Journal of Control*, 16(6):642–664, 2009.

[8] L. D. Catledge and J. E. Pitkow. Characterizing browsing strategies in the world-wide web. *Computer Networks and ISDN Systems*, 27(6):1065 – 1073, 1995.

[9] R. Cooley, B. Mobasher, and J. Srivastava. Data Preparation for Mining World Wide Web Browsing Patterns. *KNOWLEDGE AND INFORMATION SYSTEMS*, 1:5–32, 1999.

[10] T. Glatard, J. Montagnat, and P. X. Optimizing jobs timeouts on clusters and production grids. In *CCGrid07, IEEE*, pages 100 – 107, May 2007.

[11] S. Irani, S. Shukla, and R. Gupta. Online strategies for dynamic power management in systems with multiple power-saving states. *ACM Trans. Embed. Comput. Syst.*, 2(3):325–346, 2003.

[12] R. Jain. Divergence of timeout algorithms for packet retransmissions. In *5th Ann. Intl. Phoenix Conf. on Computers and Communications*, pages 174–179, March 1986.

[13] A. Kebarighotbi and C. G. Cassandras. Revisiting the optimality of cμ rule with stochastic flow models. *48th IEEE Conf. on Decision and Control*, pages 2304–2309, 2009.

[14] A. Kebarighotbi and C. G. Cassandras. *Timeout Control in Distributed Systems Using Perturbation Analysis*. `http://vita.bu.edu/akghotbi/CDC2011.pdf`, 2011.

[15] A. Kebarighotbi and C. G. Cassandras. Optimal scheduling of parallel queues using stochastic flow models. *Journal of Discrete-Event Dynamic Systems*, pages 1–30, May 2011.

[16] A. Kesselman and Y. Mansour. Optimizing TCP retransmission timeout. In P. Lorenz and P. Dini, editors, *Networking - ICN 2005*, volume 3421 of *Lecture Notes in Computer Science*, pages 133–140. Springer Berlin, 2005.

[17] L. Libman and A. Orda. Optimal retrial and timeout strategies for accessing network resources. *Networking, IEEE/ACM Tran. on*, 10(4):551 – 564, Aug. 2002.

[18] C. G. Panayiotou and C. G. Cassandras. IPA for delay threshold violation using stochastic fluid models. In *Proc. Intl. Workshop on Discrete Event Systems*, pages 162–167, May 2008.

[19] J. Peralta, P. Anussornnitisarn, and S. Nof. Analysis of a time-out protocol and its applications in a single server environment. *Intl. Journal of Computer Integrated Manufacturing*, 16(1):1, 2003.

[20] I. Psaras and V. Tsaoussidis. Why TCP timers (still) don't work well. *Comput. Netw.*, 51(8):2033–2048, 2007.

[21] P. Rong. Determining the optimal timeout values for a power-managed system based on the theory of markovian processes: offline and online algorithms. In *Design, Automation and Test in Europe*, pages 1128–1133, 2006.

[22] G. Sun, C. G. Cassandras, and C. G. Panayiotou. Perturbation analysis and optimizatin of stochastic flow networks. *IEEE Trans. on Automatic Control*, AC-49(12):2113–2128, 2004.

[23] G. Sun, C. G. Cassandras, and C. G. Panayiotou. Perturbation analysis of multiclass stochastic fluid models. *J. of Discrete Event Dynamic Systems*, 14(3):267–307, 2004.

[24] Y. Wardi, R. Adams, and B. Melamed. A unified approach to infinitesimal perturbation analysis in stochastic flow models: the single-stage case. *IEEE Trans. on Automatic Control*, 55(1):89–103, 2009.

[25] Y. Wardi and G. Riley. IPA for spillover volume in a fluid queue with retransmissions. In *Decision and Control, 2004. CDC. 43rd IEEE Conf. on*, volume 4, pages 3756 – 3761 Vol.4, dec. 2004.

[26] C. Yao. *Perturbation Analysis, Optimization and Resource Contention Games in Stochastic Hybrid Systems - Ch.6*. PhD thesis, Boston University, 2010.

[27] H. Yu and C. G. Cassandras. Perturbation analysis and feedback control of communication networks using stochastic hybrid models. *J. of Nonlinear Analysis*, 65(6):1251–1280, 2006.

[28] L. Zhang. Why TCP timers don't work well. In *Proc. of the ACM SIGCOMM Conf. on Communications architectures and protocols*, pages 397–405, Stowe, Vermont, USA, 1986.