Energy-Optimal Sampling of Edge-Based Feedback Systems

Vishnu Narayanan Moothedath
Information Science and Engineering
EECS, KTH Royal Institute of Technology
Stockholm, Sweden
vnmo@kth.se

Jaya Prakash Champati
IMDEA Networks Institute
Madrid, Spain
jaya.champati@imdea.org

James Gross
Information Science and Engineering
EECS, KTH Royal Institute of Technology
Stockholm, Sweden
jamesgr@kth.se

***Abstract*—We study a problem of optimizing the sampling interval in an edge-based feedback system, where sensor samples are offloaded to a back-end server which process them and generates a feedback that is fed-back to a user. Sampling the system at maximum frequency results in the detection of events of interest with minimum delay but incurs higher energy costs due to the communication and processing of some redundant samples. On the other hand, lower sampling frequency results in a higher delay in detecting an event of interest thus increasing the idle energy usage and degrading the quality of experience. We propose a method to quantify this trade-off and compute the optimal sampling interval, and use simulation to demonstrate the energy savings.**

***Index Terms*—Energy conservation, optimal sampling, mobile edge computing, feedback system, event detection**

I. Introduction

With the advent of next-generation mobile networks such as 5G Release 15 and 16, there is an increasing interest in realizing various real-time services and applications. Perhaps most prominently, this materializes with the Release 16 features of URLLC (ultra-reliable low latency communication) targeting sub-millisecond end-to-end delays primarily for industrial automation applications. However, in addition to these extreme use cases, a plethora of new applications are arising that all process states of reality and accurately provide feedback either to devices or humans. Examples of such feedback systems with low latency requirements are human-in-the-loop applications like augmented reality, wearable cognitive assistants (WCA) or ambient safety. Also, in the domain of cyber-physical systems (CPS), such applications are prominent, for example, in the context of automated video surveillance or distributed control systems. All these applications have in common that feedback depends on state capture and timely processing, whereas essential state changes are random events and hence an efficient operation of the application becomes a central aspect of the system. This is even more emphasized by the recent trend to place most of the processing logic of such feedback systems with edge computing facilities, leveraging supposedly ubiquitous real-time compute capabilities with the additional costs of offloading compute tasks (in terms of communication delays and energy consumption).

In this paper, we study approaches that enable capturing the relevant system changes in edge-based feedback systems while striking a balance with the total energy consumption. We consider feedback systems that monitor a process (or human activities) via sampling, while only reacting to a sub-set of samples, referred to as essential events, that lead to a system change – for instance, a new augmentation towards a human user, an alarm in a surveillance system, or an actuation in a general CPS set-up. After an essential event is captured and processed (including the generation of feedback), the feedback system transits to the next state where it starts monitoring for an essential next event to happen.

Sampling in practice is done at the highest frequency (which is dictated by the system capabilities), in order to ensure rapid detection of these essential events. However, it leads to the capture of unimportant samples of the process being observed, thereby wasting system resources in terms of energy, communication bandwidth, and computing cycles. We are interested in mathematically characterizing this trade-off and studying the implied consequences in the system design. We argue that the optimum sampling frequency is not always at the lowest end of the allowed range.

Some of the earliest ideas on detecting the relevant system changes come from control systems with applications in observable models [1], [2]. This idea is revisited in [3] where the authors look for stochastic changes in the system with an aim of quickest detection, but the resultant energy expenses are not considered. Sampling strategies, both periodic- and event-triggered, have also been extensively studied in control theory, e.g., see [4]–[6]. However, these works consider different systems (linear or non-linear) and aim to minimize control costs, e.g., Linear Quadratic Gaussian (LQG) cost, squared error distortion, ensuring stability etc.

A widely studied method to save energy is offloading the sensor data [7], [8] and the latency compounded on large number of samples by this offloading can be reduced by making offloading decisions for the samples [9]. This includes binary decisions [10]–[15], partial offloading decisions [16]–[19], and stochastic decisions [20]. However, offloading decisions tend to concentrate only on the sensor side and may take a toll on the total energy consumption. In contrast to all these works, our approach saves energy by optimizing the sampling mechanism in the first place, thereby reducing the amount of offloaded data. Instead of shifting the energy usage to the edge

device, our solution can account for the energy expended at these devices as well. In [21], though the authors designed a sampling strategy to minimize detection delay in a Markov source, there was no consideration of feedback, processing, communication, or offloading.

The paper is organised as follows. In Section. II, we discuss the general system model and the design choices. In Section III, we analyse the generalised solution along with an example to illustrate the system behaviour. We discuss the numerical results in Section IV and conclude in Section V.

II. System Model

Consider a feedback system provided with a mobile terminal (simply terminal) that samples a process and send the samples to a back-end server (simply back-end). The back-end processes these samples and (only) upon detecting an essential event, sends a feedback to the user through the terminal. For example, in a WCA system studied in [22], [23], the user is assigned to complete a set of tasks. Here, essential events correspond to the completion of each task. The camera at the terminal take snapshots of the user activity and send them to the back-end for image processing. The samples that lead to the detection of a task completion –referred to as successful samples– generates a feedback containing the instructions for the next task, and all other samples are discarded. After finishing a task, the user enters an intermediate wait period. The next sample drawn at or after this point indicates a task completion at the back-end's processor, and cause a feedback to the user. The total time from the start of one task to that of the next task is referred to as the state of the system. It consists of the time to complete the task and the following wait period. The terminal can sample the process, transmit it to the back-end and can receive a feedback whenever available. The back-end on the other hand is capable of receiving the sample, processing it and transmitting the feedback after an essential event detection. At all other times, the terminal and back-end go to their respective idle mode to save energy.

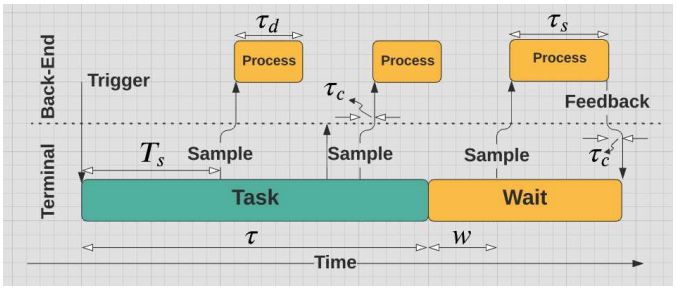


Fig. 1: System model illustrating the task, wait, communication and processing times

In this paper, we use this example to model the system. Though the back-end can possibly serve and manage multiple terminals, we restrict our study to the back-end's interaction with a single one. The timing diagram for an arbitrary task considered is given in Fig.1. We refer to the duration τ represented by the random variable $\mathcal{T}$ as the task completion time or simply, task time. Sampling the system for detecting the task completion is governed by a periodic sampling policy $\mathcal{S}$ which results in s samples being taken represented by the random variable S. This includes both the discarded samples taken during the task time as well as the final sample that leads to detecting the task completion. The duration of the total wait period that the user encounter after this task also is a random variable. This consists of the (potential) random lag w (referred to as "wait" and denoted by the random variable W) until the next sample is drawn after the task is completed along with a processing and two-way communication delay. Note that only the processing and communication delay corresponding to the final sample contributes to the wait time.

In this work, we assume that the transmit power and receive power is the same at both the terminal and back-end and we refer to it simply as communication power denoted by P_c. Let P_p be processing power at the back-end. Also, let the communication delay in either direction be τ_c and the processing delay at the back-end for the discarded and successful samples be τ_d and τ_s, respectively. We discuss the motivation of such a choice for processing delay soon. We assume that the sum of these delays is always less than the time it takes till the next sample. That is, processing and communication are always assumed to be completed before the next sampling instance. When not performing a transmission, reception, or processing, both the terminal and back-end go to their respective idle mode which demands their own idle power. The total time within a state, during which the terminal or back-end is in an idle mode is referred to as the idle time. Note that, more often than not, the values of idle time, denoted by τ_0, and idle power, denoted by P_0, are possibly different at the terminal and back-end. Hence, we make use of (t) and (b) in the superscripts to represent the terminal and the back-end, respectively. Therefore, $P_0^{(t)}$ and $P_0^{(b)}$ correspond to the power usage and $\tau_0^{(t)}$ and $\tau_0^{(b)}$ corresponds to the idle time at the terminal and back-end, respectively. The notations used and their meanings are reiterated in TABLE I for readability. We can calculate the idle times in terms of other time parameters as follows:

$$\tau_0^{(t)} = \tau + w + \tau_s - (s-1)\tau_c \quad (1)$$

$$\tau_0^{(b)} = \tau + w - (s-1)(\tau_c + \tau_d) \quad (2)$$

T_s	sampling interval	s	number of samples
τ	task time	w	wait time
τ_c	communication delay	P_c	communication power
τ_d, τ_s	processing delay	P_p	processing power
$(\cdot)^{(t)}$	at the terminal	$(\cdot)^{(b)}$	at the back-end server
τ_0	idle time	P_0	idle power
E	energy	ϵ	energy penalty
$\mathbb{E}[\cdot]$	expectation	$F_X(\cdot)$	CDF of X

TABLE I: Table of notations.

Apart from degrading the responsiveness of the system [23], a longer wait adds to the idle time thereby creating additional power overhead. It is thus desirable to minimize the expected wait $\mathbb{E}[W]$ through the choice of an aggressive

sampling policy $\mathcal{S}$. On the other hand, such a sampling policy leads to unnecessary samples taken before the task completion thereby making it undesirable from a different perspective. Communication and processing of these samples, which will be discarded eventually, warrants additional energy usage. Hence, it is desirable to minimize the expected number of samples $\mathbb{E}[S]$ by the use of relaxed sampling. It is this contrasting behaviour of W and S that motivates this paper, where we aim to find out the optimum sampling policy which minimizes the expected energy consumption. If E represents the energy requirement for the state, from (2), we get

$$\begin{aligned} E^{(t)} &= (s+1)\tau_c P_c + \tau_0^{(t)} P_0^{(t)} \\ &= s\tau_c(P_c - P_0^{(t)}) + wP_0^{(t)} + (\tau + \tau_c + \tau_s)P_0^{(t)} + \tau_c P_c \end{aligned} \quad (3)$$

$$\begin{aligned} E^{(b)} &= (s+1)\tau_c P_c + \big((s-1)\tau_d + \tau_s\big)P_p + \tau_0^{(b)} P_0^{(b)} \\ &= s\tau_c(P_c - P_0^{(b)}) + s\tau_d(P_p - P_0^{(b)}) + wP_0^{(b)} \\ &\quad + \tau P_0^{(b)} + \tau_c P_c + (\tau_s - \tau_d)P_p + (\tau_c + \tau_d)P_0^{(b)} \end{aligned} \quad (4)$$

1) Design choices: In this paper, we assume that the processing delay can take only two values τ_d and τ_s for the discarded and successful samples, respectively. This choice is motivated by the fact that the processing is often carried out with the help of a (series of) tests, like the image processing algorithm used in the example used for our modelling. Further processing is often initiated to make use of a successful sample and/or to compute the next task. In such situations, the processing time is generally governed by the completeness of the task and the successful samples tend to require a much larger processing time compared to any of the discarded samples. Hence an assumption of a two-valued processing time distribution is well justified.

2) Problem Statement: With this understanding, the problem that we aim to solve boils down to finding the sampling interval under a periodic sampling policy which will minimize the expected energy consumption for the given task time statistics. Recall the expressions (3) and (4). The terms except those containing the number of samples s or the wait w are either constants or have constant expectations for a fixed task time distribution. Hence those terms becomes irrelevant in the optimization where we minimize the expected energy. Let $\epsilon(T_s)$ be the component of the total energy which is relevant for the optimization and let us call it *energy penalty*. That is,

$$\epsilon^{(t)}(T_s) = \alpha^{(t)}\mathbb{E}[S] + \beta^{(t)}\mathbb{E}[W] \quad (5)$$

$$\epsilon^{(b)}(T_s) = \alpha^{(b)}\mathbb{E}[S] + \beta^{(b)}\mathbb{E}[W] \quad (6)$$

$$\begin{aligned} &\alpha^{(t)} = \tau_c(P_c - P_0^{(t)}),\ \beta^{(t)} = P_0^{(t)}, \\ &\alpha^{(b)} = \tau_c(P_c - P_0^{(b)}) + \tau_d(P_p - P_0^{(b)}),\text{ and } \beta^{(b)} = P_0^{(b)} \end{aligned} \quad (7)$$

The basic structure of the energy penalty is the same both at the terminal and back-end. Consider the general optimization problem below to find the optimum sampling interval T_S^*

$$T_s^* = \underset{T_s}{\text{Min}}\ \epsilon(T_s) = \underset{T_s}{\text{Min}}\ \big(\alpha\mathbb{E}[S] + \beta\mathbb{E}[W]\big) \quad (8)$$

Here, β represents the idle power with $\beta\mathbb{E}[W]$ corresponding to the additional energy expended for waiting, and α represents the energy wasted per discarded sample due to the additional communication and/or processing. Once solved, one can always use this to find the optimum sampling interval from a terminal perspective, back-end perspective or from a combined energy perspective only by properly choosing the coefficients α and β from (7). One can even assign different costs for energy at the terminal and back-end. For instance, for a system where energy at the terminal costs as much as that of the back-end, the coefficients will be $\alpha = \alpha^{(t)} + \alpha^{(b)}$ and $\beta = \beta^{(t)} + \beta^{(b)}$.

III. Analysis

In this section, we find the optimum periodic sampling interval by minimizing $\epsilon(T_s)$ in (8). The optimization technique can be specific to the task time distribution and we illustrate this using an example where the task times are assumed to be exponentially distributed. Before that, we start with a generalised approach where we find $\mathbb{E}[W]$ and $\mathbb{E}[S]$.

A. General Task Time Distribution

First, we derive the expectation of the number of samples s and the wait w, which together constitutes $\epsilon(T_s)$.

1) Number of samples: Recall that s denotes the number of samples taken for a task and S denotes the corresponding random variable. We have,

$$\begin{aligned} \mathbb{P}(s=k) &= \mathbb{P}\big(\lceil \tau/T_s \rceil = k\big), \forall k \geq 1 \\ &= \mathbb{P}(k-1 < \tau/T_s \leq k) \\ &= F_{\mathcal{T}}\big(kT_s\big) - F_{\mathcal{T}}\big((k-1)T_s\big) \\ \Rightarrow \mathbb{E}[S] &= \sum_{k=1}^{\infty} k\Big(F_{\mathcal{T}}\big(kT_s\big) - F_{\mathcal{T}}\big((k-1)T_s\big)\Big). \end{aligned} \quad (9)$$

2) Wait: Recall that w denotes the time waited from the task completion to the very next sample and W denotes the corresponding random variable. We have a fixed set of sampling instances governed by T_s. The CDF of wait penalty $F_W(w)$ can thus be obtained by taking the probability of the task time to fall at most w short of any sampling instance. Even though task times are finite in practice, we consider they can be arbitrarily large for the sake of generalized analysis. As a result, a successful sample can be located anywhere from the first sampling instance to possibly infinity. Thus,

$$\begin{aligned} F_W(w) &= \sum_{k=1}^{\infty} \mathbb{P}\big(kT_s - w < \tau \leq kT_s\big) \\ &= \sum_{k=1}^{\infty} \big(F_{\mathcal{T}}(kT_s) - F_{\mathcal{T}}(kT_s - w)\big). \end{aligned} \quad (10)$$

Since W is a non-negative random variable, we have,

$$\begin{aligned} \mathbb{E}[W] &= \int_0^{\infty} \big(1 - F_W(w)\big)dw \\ &= \int_0^{T_s} \Big(1 - \sum_{k=1}^{\infty} \big(F_{\mathcal{T}}(kT_s) - F_{\mathcal{T}}(kT_s - w)\big)\Big)dw \end{aligned}$$

Since the sum and limits are finite and the summand is non-negative,

$$= T_s - \sum_{k=1}^{\infty} \int_0^{T_s} \big(F_{\mathcal{T}}(kT_s) - F_{\mathcal{T}}(kT_s - w)\big)dw \quad (11)$$

We can calculate the energy penalty using (8), (11) and (9). Notice that $\mathbb{E}[S]$ decreases and $\mathbb{E}[W]$ increases with T_s. This opposing behaviour of the penalties generate a minima in the weighted sum $\epsilon(T_s)$ (with weights α and β) at T_s^*-the optimum sampling interval. Depending on the distribution of task times and the resultant energy penalty expression, T_s^* can be computed using known optimization techniques or numerical solvers [24].

B. Exponential Task Time Distribution

In this subsection, we look into such a feedback system where the task completion times are exponentially distributed.

Lemma 1. *For exponentially distributed task times with mean $1/\lambda$, the energy penalty $\epsilon(T_s)$ imparted by a periodic sampling policy with a period T_s is given by*

$$\epsilon(T_s) = \frac{\alpha\lambda + \beta\big(e^{-\lambda T_s} + \lambda T_s - 1\big)}{\lambda(1 - e^{-\lambda T_s})} \quad (12)$$

Proof. We have $F_{\mathcal{T}}(\tau) = 1 - e^{-\lambda\tau}$. From (9), the expected number of samples can be computed as,

$$\begin{aligned}\mathbb{E}[S] &= \sum_{k=1}^{\infty} k\Big((1 - e^{-\lambda k T_s}) - (1 - e^{-\lambda(k-1)T_s})\Big) \\ &= (1 - e^{-\lambda T_s})^{-1} \end{aligned} \quad (13)$$

Substituting $F_{\mathcal{T}}(\tau)$ in (10),

$$\begin{aligned} F_W(w) &= \sum_{k=1}^{\infty} \big((1 - e^{-\lambda k T_s}) - (1 - e^{-\lambda(kT_s - w)})\big) \\ &= \sum_{k=1}^{\infty} e^{-\lambda k T_s}(e^{\lambda w} - 1) = \frac{e^{\lambda w} - 1}{e^{\lambda T_s} - 1} \\ \Rightarrow f_W(w) &= \frac{\lambda e^{\lambda w}}{e^{\lambda T_s} - 1} \\ \Rightarrow \mathbb{E}[W] &= \int_0^{T_s} \frac{\lambda w e^{\lambda w}}{e^{\lambda T_s} - 1} dw \\ &= \frac{1}{\lambda(e^{\lambda T_s} - 1)} \int_0^{\lambda T_s} x e^x dx \\ &= \frac{e^{-\lambda T_s} + \lambda T_s - 1}{\lambda(1 - e^{-\lambda T_s})} \end{aligned} \quad (14)$$

From (13) and (14), $\epsilon(T_s)$ for using a particular sampling interval T_s, can be expressed using (8) as

$$\epsilon(T_s) = \frac{\alpha\lambda + \beta\big(e^{-\lambda T_s} + \lambda T_s - 1\big)}{\lambda(1 - e^{-\lambda T_s})}$$

□

Lemma 2. *The energy penalty $\epsilon(T_s)$ is convex in T_s.*

Proof. In the following, for simplicity in presentation we drop (T_s). Note that $\beta > 0$ (cf.(7)).

$$\begin{aligned} \epsilon' &:= \frac{d\epsilon}{dT_s} = \frac{\beta\big(1 - e^{-\lambda T_s}(\lambda T_s + \frac{\alpha}{\beta}\lambda + 1)\big)}{(1 - e^{-\lambda T_s})^2} \\ \epsilon'' &:= \frac{d^2\epsilon}{dT_s^2} = \frac{\beta\lambda e^{-\lambda T_s}\big((1 + e^{-\lambda T_s})(\lambda T_s + \frac{\alpha}{\beta}\lambda) + 2e^{-\lambda T_s} - 2\big)}{(1 - e^{-\lambda T_s})^3} \end{aligned}$$

$$\begin{aligned} \epsilon'' \geq 0 \Rightarrow \tilde{\epsilon} &:= (1 + e^{-\lambda T_s})(\lambda T_s + \tfrac{\alpha}{\beta}\lambda) + 2e^{-\lambda T_s} - 2 \geq 0 \\ \tilde{\epsilon}' &:= \frac{d\tilde{\epsilon}}{dT_s} = \lambda(1 - e^{-\lambda T_s}(\lambda T_s + \tfrac{\alpha}{\beta}\lambda + 1)) \\ \tilde{\epsilon}'' &:= \frac{d^2\tilde{\epsilon}}{dT_s^2} = \lambda^2 e^{-\lambda T_s}(\lambda T_s + \tfrac{\alpha}{\beta}\lambda) \end{aligned}$$

From the above expressions, since $\tilde{\epsilon}'' \geq 0\ \forall T_s$, we can conclude that $\tilde{\epsilon}$ is globally convex and any infimum point is it's minimum. To find this infimum:

$$\tilde{\epsilon}' = 0 \Rightarrow \lambda T_s + \tfrac{\alpha}{\beta}\lambda = e^{\lambda T_s} - 1.$$

Substituting in the above expression for $\tilde{\epsilon}$, we obtain

$$\begin{aligned} \underset{T_s>0}{Min}\{\tilde{\epsilon}\} =& (1 + e^{-\lambda T_s})(e^{\lambda T_s} - 1) + 2e^{-\lambda T_s} - 2 \\ =& e^{\lambda T_s} + e^{-\lambda T_s} - 2 \\ \geq& 0\ \forall T_s > 0 \\ \Rightarrow \tilde{\epsilon} \geq& 0\ \forall T_s > 0 \\ \Rightarrow \epsilon'' \geq& 0\ \forall T_s > 0 \end{aligned}$$

Note that T_s is always non-negative. Thus the energy penalty ϵ is convex in the region of interest. □

Proposition 1. *The optimum sampling interval T_s^* for a WCA with an exponentially distributed task times with mean $1/\lambda$ is the solution of the expression $e^{\lambda T_s^*} - \lambda T_s^* = \frac{\alpha}{\beta}\lambda + 1$.*

Proof. The proof is straightforward from Lemma. 1 and 2. We can find the optimum by equating the $\epsilon'(T_s)$ to zero.

$$\begin{aligned} \frac{d\epsilon(T_s)}{dT_s} = 0 \Rightarrow 1 - e^{-\lambda T_s}(\lambda T_s + \tfrac{\alpha}{\beta}\lambda + 1) = 0 \\ \Rightarrow e^{\lambda T_s} - \lambda T_s = \tfrac{\alpha}{\beta}\lambda + 1 \end{aligned}$$

□

This monotonically increasing convex function in a single variable T_s can be solved using well-known numerical solvers. Though the value of energy penalty differs with the values of β and α, the optimum sampling interval T_s^* only depends on their ratio. Furthermore, from (7), it can be seen that this ratio β/α does not depend on the individual power figures but only on the percentage additional power necessary for communication or processing when compared to their respective idle power requirement. In other words, for fixed λ, β/α and thus the optimum sampling interval is a function of only τ_c and $P_c/P_0^{(t)}$ at the terminal, and $\tau_c, \tau_d, P_c/P_0^{(b)}$, and $P_p/P_0^{(b)}$ at the back-end.

IV. Numerical Results

In this section, we present the numerical results to illustrate the system behaviour. We use an exponentially distributed task time with a mean of 10s and consider two characterisations as given in TABLE II. The former is motivated from a cyber-physical system (CPS) and while the latter from a video analytic system (VAS). The processing and communication powers are assumed to be lower for a CPS that has to transmit and process less amounts of data compared to a video feed. We use comparable terminals with same idle power but the remaining power figures are larger for a VAS.

	$P_0^{(t)}$	$P_0^{(b)}$	P_c	P_p	τ_c	τ_p
CPS	4mW	5mW	0.1W	30mW	1ms	8ms
VAS	4mW	8mW	0.8W	40mW	10ms	50ms

TABLE II: Parameters used for a cyber-physical system(CPS) and a video analytic system (VAS).

In Fig. 2, we show the variation of $\mathbb{E}[S]$ and $\mathbb{E}[W]$ with sampling interval T_s. These expectations depend only on the system parameters λ and T_s but not on the parameters that dictate the energy usage. Observe that $\mathbb{E}[S]$ decreases and $\mathbb{E}[W]$ increases with T_s, as discussed in Section II. After a rapid decrease for small T_s, $\mathbb{E}[S]$ converges to 1, denoting a single sample. On the other hand, $\mathbb{E}[S]$ increases with T_s approximately in a linear manner.

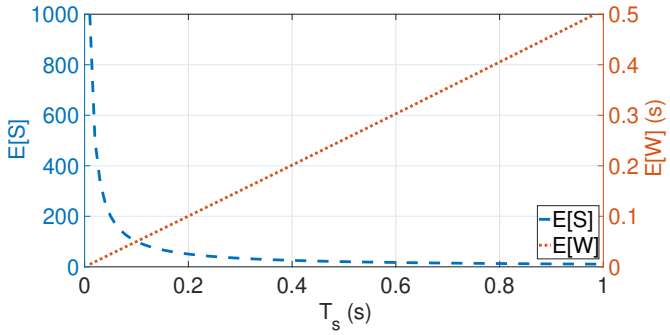
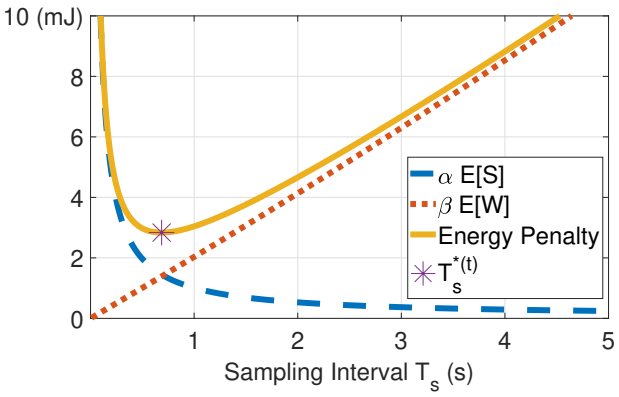
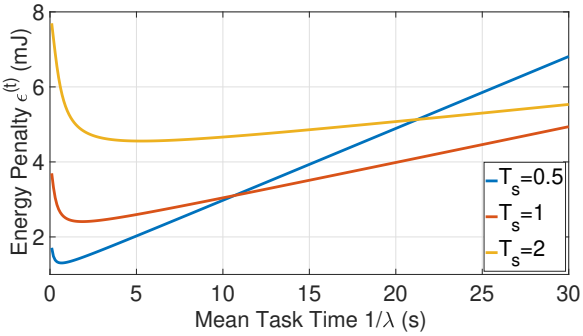


Fig. 2: Expected number of samples $\mathbb{E}[S]$ and expected wait penalty $\mathbb{E}[W]$ vs. sampling interval.

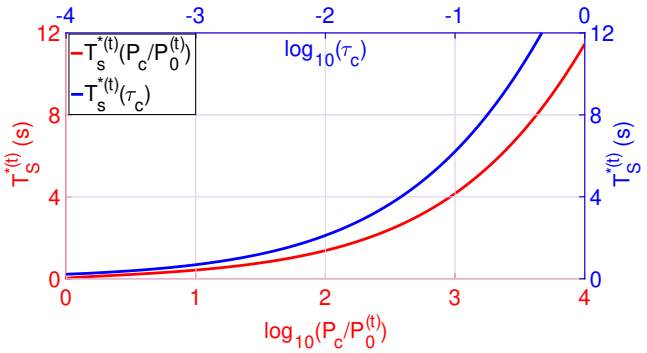
For all plots in Fig. 3, we use CPS from TABLE. II and consider only the energy penalty at the terminal side, resulting in β/α of 41.67. The energy penalty and its components are plotted against the sampling interval T_s in Fig. 3a. The asymptotic decrease of $\alpha\mathbb{E}[S]$ to α and the steady increase of $\beta\mathbb{E}[W]$ with T_s result in a minima for $\epsilon^{(t)}$ at $T_s^{*(t)} = 0.686$. This optimum sampling interval, calculated by solving (1) using a bisection search is also shown in the figure. Due to the behaviour of its significant component at a particular sampling interval, the energy penalty curve shows a rapid decrease with T_s for $T_s < T_s^{*(t)}$ followed by a linear increase for $T_s > T_s^{*(t)}$. It can thus be inferred from an energy standpoint that, a negative error in choosing the optimum periodic sampling interval is typically costlier than a positive error of the same amount. In



(a) Energy penalty and its components Vs. sampling interval.



(b) Energy penalty vs. Mean task time for fixed sampling interval.



(c) Optimum sampling interval vs. ratio of communication and idle powers $P_c/P_0^{(t)}$ and communication delay τ_c.

Fig. 3: Penalties and T_s^* at the terminal of CPS

Fig. 3b, we vary the mean task time and show its effect on $\epsilon^{(t)}$ for sampling intervals chosen around the optimum value from Fig. 3a. $\epsilon^{(t)}$ initially decreases rapidly and later increases slowly with an increase in mean task time. In Fig. 3c, we show the dependency of P_c and τ_c, where one parameter is varied and the other one is fixed at its default value. A change in P_c or τ_c, results in a proportional change in β/α, which in turn results in an inversely proportional change in $T_s^{*(t)}$. Larger communication delay or power results in larger optimum sampling interval, thereby maintaining the balance between the total idle energy and communication energy consumed.

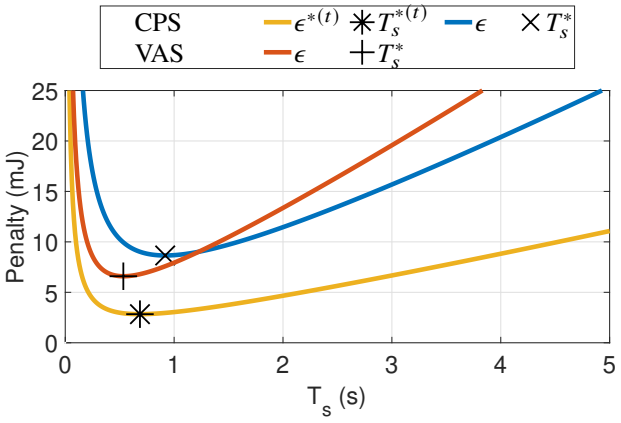


Fig. 4: Combined energy penalty of the terminal and back-end, for CPS and VAS.

In Fig. 4, we plot the combined energy penalty for CPS and VAS by considering both the back-end and the terminal. α and β are calculated as the sum of the respective coefficients at the terminal and back-end, thereby getting an β/α ratio of 23.0 and 68.6, respectively. For comparison, we have also added $\epsilon^{(t)}$ for CPS from Fig. 3a. Observing all the results presented for the given parameters, we conclude that the proposed optimum sampling interval can reduce energy consumption significantly. Furthermore, negative errors in the computation of optimum sampling interval should be particularly avoided to prevent a potential spike in energy usage.

V. Conclusion

In this paper, we considered an edge-based feedback system that captures essential events via periodic sampling. We proposed an optimization framework with which the sampling interval that minimizes the energy consumption of this feedback system can be computed. Apart from the generic approach to solve the optimization problem for an arbitrary task time distribution, we also illustrated an example feedback system with an exponentially distributed task completion time. This example is then used to illustrate the system behaviours in the numerical section where we discussed the energy savings enabled by using the computed sampling interval. We also discussed the additional energy expended as a result of a computation error and concluded that this expense is relatively steeper for a negative error.

In this work, the exponential task times serves the purpose of an initial proof of concept. In future we plan to change this to statistics inspired from analysing real-world data. We also plan to extend the system model to non-deterministic processing delays.

VI. Acknowledgement

This research has been partially funded by the VINNOVA Competence Center for Trustworthy Edge Computing Systems and Applications (TECoSA) at KTH Royal Institute of Technology.

References

[1] W. A. Shewhart, "The application of statistics as an aid in maintaining quality of a manufactured product," *Journal of the American Statistical Association*, vol. 20, no. 152, pp. 546–548, 1925. [Online]. Available: http://www.jstor.org/stable/2277170

[2] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, no. 1/2, pp. 100–115, 1954. [Online]. Available: http://www.jstor.org/stable/2333009

[3] V. V. Veeravalli and T. Banerjee, "Quickest change detection," 2012.

[4] C. Kam, S. Kompella, G. D. Nguyen, J. E. Wieselthier, and A. Ephremides, "Towards an effective age of information: Remote estimation of a markov source," in *IEEE INFOCOM WKSHPS*, April 2018, pp. 367–372.

[5] S. Feng and J.-S. Yang, "Information freshness for timely detection of status changes," *ArXiv*, vol. abs/2002.04648, 2020.

[6] Y. Inoue and T. Takine, "AoI perspective on the accuracy of monitoring systems for continuous-time markovian sources," in *IEEE INFOCOM WKSHPS*, April 2019, pp. 183–188.

[7] B. Shi, J. Yang, Z. Huang, and P. Hui, "Offloading guidelines for augmented reality applications on wearable devices," in *Proc. of ACM International Conference on Multimedia*, 2015, p. 1271–1274.

[8] R. Kemp, N. Palmer, T. Kielmann, F. Seinstra, N. Drost, J. Maassen, and H. Bal, "eyeDentify: Multimedia cyber foraging from a smartphone," in *Proc. IEEE International Symposium on Multimedia*, 2009, pp. 392–399.

[9] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[10] J. P. Champati and B. Liang, "Semi-online algorithms for computational task offloading with communication delay," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 4, pp. 1189–1201, 2017.

[11] ——, "Single restart with time stamps for parallel task processing with known and unknown processors," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 1, pp. 187–200, 2020.

[12] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE TWC*, vol. 12, no. 9, pp. 4569–4581, 2013.

[13] C. You, K. Huang, and H. Chae, "Energy efficient mobile cloud computing powered by wireless energy transfer," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1757–1771, 2016.

[14] K. Kumar and Y. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *Computer*, vol. 43, no. 4, pp. 51–56, 2010.

[15] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, "Communicating while computing: Distributed mobile cloud computing over 5G heterogeneous networks," *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 45–55, 2014.

[16] M. Zhao, J.-J. Yu, W.-T. Li, D. Liu, S. Yao, W. Feng, C. She, and T. Q. S. Quek, "Energy-aware offloading in time-sensitive networks with mobile edge computing," *CoRR*, vol. abs/2003.12719, 2020.

[17] S. E. Mahmoodi, R. N. Uma, and K. P. Subbalakshmi, "Optimal joint scheduling and cloud offloading for mobile applications," *IEEE Transactions on Cloud Computing*, vol. 7, no. 2, pp. 301–313, 2019.

[18] W. Zhang, Y. Wen, and D. O. Wu, "Collaborative task execution in mobile cloud computing under a stochastic wireless channel," *IEEE TWC*, vol. 14, no. 1, pp. 81–93, 2015.

[19] S. E. Mahmoodi, K. P. Subbalakshmi, and V. Sagar, "Cloud offloading for multi-radio enabled mobile devices," in *Proc. of 2015 IEEE ICC*, 2015, pp. 5473–5478.

[20] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE TWC*, vol. 11, no. 6, pp. 1991–1995, 2012.

[21] J. P. Champati, M. Skoglund, and J. Gross, "Detecting state transitions of a markov source: Sampling frequency and age trade-off," in *Proc. IEEE INFOCOM Workshops*, 2020, pp. 7–12.

[22] Z. Chen, W. Hu, J. Wang, S. Zhao, B. Amos, G. Wu, K. Ha, K. Elgazzar, P. Pillai, R. Klatzky, D. Siewiorek, and M. Satyanarayanan, "An empirical study of latency in an emerging class of edge computing applications for wearable cognitive assistance," in *Proc. ACM/IEEE Symposium on Edge Computing*, 2017.

[23] M. O. Muñoz, R. Klatzky, J. Wang, P. Pillai, M. Satyanarayanan, and J. Gross, "Impact of delayed response on wearable cognitive assistance," *CoRR*, vol. abs/2011.02555, 2020.

[24] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.