Data Freshness in Mixed-Memory Intermittently-Powered Systems

James Scott Broadhead and Przemysław Pawełczak

Embedded and Networked Systems, EEMCS, Delft University of Technology, The Netherlands

Email: {J.S.Broadhead, P.Pawelczak}@tudelft.nl

Abstract—Age of Information (AoI) is a key metric to understand data freshness in Internet of Things (IoT) devices. In this paper we analyse an intermittently-powered IoT sensor—with mixed-memory (volatile and non-volatile) architecture—that uses a Time-Dependent Checkpointing (TDC) scheme. We derive the average Peak Age of Information (PAoI) and average AoI of the system, and use these metrics to understand which device parameters most significantly influence performance. We go on to consider how the average PAoI of a mixed-memory system compares with entirely volatile or entirely non-volatile architecture, and also introduce an alternative TDC strategy to improve system resilience in unpredictable environmental conditions.

I. INTRODUCTION

With an increasing paradigm shift towards battery-free energy harvesting-based design in low-powered embedded systems, new methods of operation have been developed to address the inherent intermittency of available harvested energy. Current *intermittent computing techniques* [1]–[5] seek to minimise the time and energy impact of power failure by strategically checkpointing—effectively saving—the system state from Volatile Memory (VM) to Non-Volatile Memory (NVM) in mixed-memory systems [6] (such as the popular Texas Instruments MSP430 micro-controller [1, Section 3.1] [7, Section 2.1.1]). Whilst much work has been done to develop new checkpointing strategies, there is still opportunity to better describe these systems mathematically [7, Section 3.2.2], in particular from a data freshness perspective.

A relevant metric to measure data freshness is the Age of Information (AoI) [8]–[14], which will form the basis of our analysis. We seek to model an Intermittently-Powered Device (IPD), implementing Time-Dependent Checkpointing (TDC) [15, Fig. 3], [16] where the VM system state is saved to NVM after a certain number of clock ticks have passed, and observe how fundamental system parameters (failure rate, checkpointing overhead, etc.) affect the freshness of locally sensed data. We also look to compare the mixed-memory architecture of our IPD with single-type VM and single-type NVM memory structures. We go on to introduce an alternative TDC scheme, Split-Frequency Checkpointing (SFC), and consider how this can improve system resilience in unpredictable environmental conditions.

Our work builds on [17] by accounting for the mixed-memory nature of many battery-free devices and the checkpointing schemes used to move data between memory types. To the best of our knowledge, this is the first evaluation and comparison of AoI in IPDs with mixed-memory architecture.
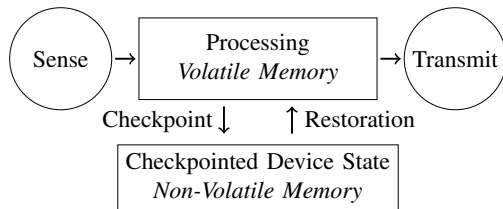

Figure 1. System model for an IPD that checkpoints its internal system state to protect from data corruption. The device performs sensing, on-board processing (that takes P_i clock ticks to complete), and transmission. Sensing and transmission occur instantaneously. Data stored in VM is checkpointed to NVM taking D_i clock ticks, and system state restoration from NVM to VM takes V_i clock ticks. The system suffers frequent power failures.

The efficiency of TDC has been considered in other areas of research, notably for distributed stream processing [18] (which looked to minimise system utilization) and also in High Performance Computing (HPC) applications [19] (which used *wall-clock length* as the objective). However, this form of checkpointing has not been analysed for transiently-operating embedded devices, or with system freshness as the core tenet of consideration—which forms the premise of our work.

In this paper we identify the average Peak Age of Information (PAoI) and average AoI for an IPD that uses TDC. These results allow us to better understand the role of mixed-memory architecture in IPDs and how the inter-checkpointing time can be best adjusted to minimise AoI. We also show that mixed-memory architecture can improve the system freshness of an IPD compared with single-type VM, however it cannot surpass entirely NVM architecture. We further show that SFC can improve system performance in unpredictable environmental conditions compared to inappropriately assigned single-frequency checkpoint intervals.

II. SYSTEM MODEL

We consider a communication device, presented in Fig. 1, which is powered by an intermittent energy source (sun, vibrations, temperature gradient, etc.) and consequently suffers frequent power failure. The device has mixed-memory architecture (VM and NVM) and checkpoints the system state (processor registers, hardware registers, main memory, etc.) from VM to NVM after a fixed number of clock ticks, where clock ticks act as a base unit for the system's on-board clock. The sensed data is then processed and transmitted, e.g. wirelessly through a low-powered LED [7], to a central collecting unit.

Packet Generation/Sensing. Packets containing data from an on-board sensor are produced as required in a *generate-at-*

will type policy—where sensing only occurs when processing of the preceding packet is complete—as considered in [17]. We assume that sensing is instantaneous and that this data is then immediately processed, from which a packet is created. Packets are not dropped due to power failure, since the last computation state can always be restored from NVM to VM.

System Operation. *Data processing* occurs in the volatile memory of Fig. 1 and encompasses a number of possible steps; such as peripheral control, filtering, and packet framing. To reduce unnecessary system complexity we have considered all processing as one stage that takes P_i clock ticks to complete. The time between data sensing and packet transmission, the *completion time*, is S_i clock ticks and the time between two sequential packet transmissions, the *inter-completion time*, is Y_i clock ticks, where the idle time between a packet transmission and the next generated packet is I_i clock ticks.

System Failure. IPDs suffer frequent power failures due to a lack of available harvested energy. We do not explicitly consider energy as part of our system model, as in [9, Section V], [17], [20], rather assuming that the depleted energy will cause a number of random power failures. An amount of processing time $L_{i,j}$ clock ticks, where the i represents the overall cycle number and j the fail number within the cycle, will be wasted for each fail (since the updated system state including this processing has not been checkpointed from VM to NVM before failure). Our system will also be inactive for a period of $R_{i,j}$ clock ticks after failure. During power failure no new data is generated since the device cannot perform sensing. Once power is restored the system takes $V_{i,j}$ clock ticks to restore the last checkpointed system state from NVM to VM. For simplicity of analysis we assume that $V_{i,1} = V_{i,j} = V$ for all j. In practice this term would likely be fixed by design.

Checkpointing Strategy. The system will save the current device state stored in VM to NVM with a pre-determined regularity. This inter-checkpoint time is given by $K_{i,n}$ clock ticks where n is the checkpoint number within cycle i. The action of checkpointing will also take a fixed amount of $D_{i,n}$ clock ticks to complete. Here, for simplicity of analysis, we assume that $D_{i,1} = D_{i,n} = D$ for all n. This is consistent with, for example, [15, Fig. 3]. We also assume that P_i is a multiple of the inter-checkpointing time.

Transmission. Our model considers the transmissions of packets to be instantaneous and occurring after a final checkpoint.

III. AGE OF INFORMATION BACKGROUND

Let us now re-introduce several previously derived results that form the basis of our analysis. Canonically we define AoI as [17, Eq. (1)] [9, Section II]

$$\Delta(t) = t - u(t), \tag{1}$$

where $\Delta(t)$ is the AoI of data sensed by the device, t is the current time, and $u(t)$ is the time stamp of the last completed packet. The AoI time evolution for our proposed system model is depicted in Fig. 2. Using this graphical representation we
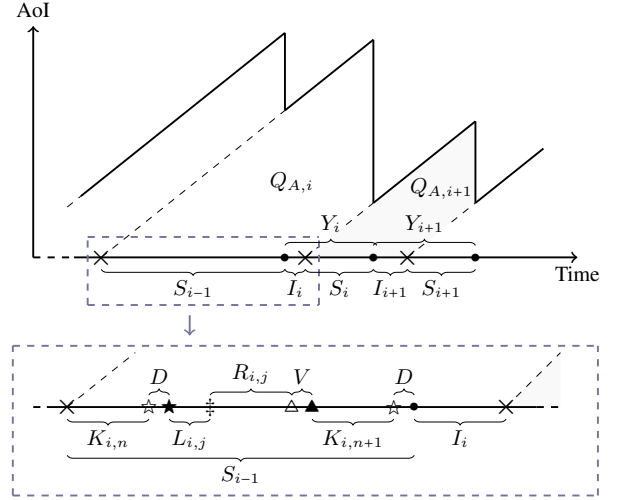

Figure 2. AoI evolution for an IPD that checkpoints its state with a fixed regularity. Symbol notation: $\times$ denotes the start of sensing, $\ddagger$ denotes device failure, $\bullet$ denotes the end of the final checkpoint and instantaneous packet transmission, $\star$ (open star) denotes the start of checkpointing from VM to NVM, $\bigstar$ denotes the end of checkpointing from VM to NVM, $\triangle$ denotes start of restoration from NVM to VM, and $\blacktriangle$ denotes the end of restoration from NVM to VM. All variables are defined in Section II.

can calculate the expectation of (1) as follows. Let us denote a trapezium $Q_{A,i}$, whose area is given by

$$Q_{A,i} = \frac{1}{2}\left((S_{i-1} + Y_i)^2 - S_i^2\right). \tag{2}$$

This is the isosceles triangle created by $S_{i-1} + Y_i$ minus the smaller isosceles triangle with base S_i. The area $Q_{A,i+1}$ is highlighted in Fig. 2 as an example. Given that S_{i-1} and Y_i are independent and S_{i-1} and S_i are identically distributed, as considered in [17, Eq. (5)], the expectation of (2) becomes

$$\mathbb{E}[Q_{A,i}] = \frac{1}{2}\mathbb{E}[Y_i^2] + \mathbb{E}[S_{i-1}]E[Y_i]. \tag{3}$$

As argued in [17, Eq. (6)], and in concordance with the geometry presented in Fig. 2, the average AoI is

$$\mathbb{E}[\Delta] = \lambda\,\mathbb{E}[Q_{A,i}], \tag{4}$$

where λ is the interarrival time of packets to the device. Since our system model considers packet generation, rather than packet arrival, the distribution of λ will be predicated on the distribution of time between sensing (which is equivalent to the distribution of time between transmission) hence $\lambda = \frac{1}{E[Y]}$. Here we introduce Y and S for the *inter-completion time* and *completion time*, respectively, when taking their expected values, $\mathbb{E}[Y]$ and $\mathbb{E}[S]$, as time tends to infinity. This is possible due to the ergodicity of the system. As such (4) simplifies to [17, Eq. (6)]

$$\mathbb{E}[\Delta] = \frac{\mathbb{E}[Y^2]}{2\,\mathbb{E}[Y]} + \mathbb{E}[S]. \tag{5}$$

The average PAoI, a more manageable measure of data freshness compared to average AoI, is given by [17, Eq. (7)] [9, Eq. (8)]

$$\mathbb{E}[\Delta^{\text{Peak}}] = \mathbb{E}[Y] + \mathbb{E}[S]. \tag{6}$$

IV. COMPLETION AND INTER-COMPLETION TIME

Given (5) and (6) it is imperative that we find expressions for Y_i and S_i from which we can calculate their expected values, $\mathbb{E}[Y]$ and $\mathbb{E}[S]$, over many cycles. From Fig. 2 we see that the *inter-completion time* for cycle i is

$$Y_i = \underbrace{I_i}_{\text{Idle time}} + \underbrace{\sum_{j=1}^{f} (L_{i,j} + R_{i,j} + V)}_{\text{Events associated with failure}}$$

$$+ \underbrace{\sum_{n=1}^{h} K_{i,n}}_{\text{Processing} \triangleq P_i} + \underbrace{Dh}_{\text{Checkpointing}}, \qquad (7)$$

where f and h are the number of system fails and the number of successfully performed checkpoints during the period of time Y_i, respectively. The time between data being sensed and a packet being transmitted, the *completion time*, for cycle i is

$$S_i = Y_i - I_i. \qquad (8)$$

V. EXPECTATION OF COMPLETION AND INTER-COMPLETION TIME

To identify the expected values of AoI in (5) and (6), and hence understand the freshness of data produced by the modelled IPD, we must first find the expected values of (7) and (8) over many cycles. We assume that all variables in (7) and (8) have well-defined means and known variance. Due to the assumed ergodicity of the system we can also simplify notation when taking the expected values of variables (for example, the expectation of I_i is $\mathbb{E}[I]$ as t tends to infinity). By following the approach of [17, Eq. (10)], we use Wald's identity [21, Proposition 1.8.1] to find that the expected values of *inter-completion time* and *completion time* are

$$\mathbb{E}[Y] = \mathbb{E}[I] + \mathbb{E}[f] (\mathbb{E}[L] + \mathbb{E}[R] + V)$$
$$+ \mathbb{E}[h] \mathbb{E}[K] + D \mathbb{E}[h], \qquad (9)$$
$$\mathbb{E}[S] = \mathbb{E}[Y] - \mathbb{E}[I], \qquad (10)$$

respectively. By definition, the amount of wasted processing per failure will take a value $L_{i,j} \in [1 .. K_{i,\theta}+D]$ clock ticks where $L_{i,j} \in \mathbb{N}^+$ and θ is the checkpoint number of the processing cycle started but interrupted in $L_{i,j}$. For instance in Fig. 2, $\theta = n+1$ since the $K_{i,n+1}$ cycle was initially started in $L_{i,j}$, however could not finish before failure. We also assume that all possible values of $L_{i,j}$ are equally likely. Therefore, based on these assumptions, the expectation of wasted processing time per failure is

$$\mathbb{E}[L] = \frac{\mathbb{E}[K] + D + 1}{2}. \qquad (11)$$

Given the definition of processing time P_i, given in the third summand of (7), its expected value will be

$$\mathbb{E}[P] = \mathbb{E}[h] \mathbb{E}[K]. \qquad (12)$$

Substituting (11) and (12) into (9) yields

$$\mathbb{E}[Y] = \mathbb{E}[f] \left(C_1 + \frac{\mathbb{E}[P]}{2 \mathbb{E}[h]} \right) + C_2 + D \mathbb{E}[h], \qquad (13)$$

where $C_1 = \mathbb{E}[R] + V + \frac{D+1}{2}$, and $C_2 = \mathbb{E}[I] + \mathbb{E}[P]$ are defined for compactness of presentation.

VI. EXPECTATION OF AVERAGE PAoI

Given (6) and expressions (10) and (13) the average PAoI for our mixed-memory IPD is therefore

$$\mathbb{E}[\Delta^{\text{Peak}}]_{\text{MM}} = 2 \mathbb{E}[f] \left(C_1 + \frac{\mathbb{E}[P]}{2 \mathbb{E}[h]} \right) + C_2$$
$$+ \mathbb{E}[P] + 2D \mathbb{E}[h]. \qquad (14)$$

We can see from this expression the clear conflict between over and under checkpointing. If failure occurs, which is a core phenomenon associated with an IPD, then an increase in the checkpointing frequency (meaning an increase in $\mathbb{E}[h]$) increases $\mathbb{E}[\Delta^{\text{Peak}}]_{\text{MM}}$ through the final $2D \mathbb{E}[h]$ term—i.e. the overhead associated with checkpointing. In contrast, an increase in $\mathbb{E}[h]$ also reduces the $\frac{\mathbb{E}[P]}{2 \mathbb{E}[h]}$ term, since more frequent checkpointing reduces the expected wasted processing time per power failure. From (14) we also observe that the constant C_1 (and hence the constants V and D) increase the average PAoI, so these should be minimised by systems designers. Further, increases in $\mathbb{E}[I]$ and $\mathbb{E}[R]$ will also increase $\mathbb{E}[\Delta^{\text{Peak}}]_{\text{MM}}$, however we consider these to be outside the control of the designer.

Minimising the Average PAoI. Due to the conflict of over and under checkpointing we can find a value of $\mathbb{E}[h]$ to minimise the average PAoI as follows. Taking the derivative of (14), with respect to $\mathbb{E}[h]$ we have

$$\frac{d \mathbb{E}[\Delta^{\text{Peak}}]_{\text{MM}}}{d \mathbb{E}[h]} = 2D - \frac{\mathbb{E}[f] \mathbb{E}[P]}{\mathbb{E}[h]^2}. \qquad (15)$$

Then, letting (15) equal to zero and solving for $\mathbb{E}[h]$, we find that (14) is minimised when

$$\mathbb{E}[h] = \sqrt{\frac{\mathbb{E}[f] \mathbb{E}[P]}{2D}}. \qquad (16)$$

This expression is comparable to [19, Eq. (7)] and shows that checkpoint optimisation is not inherently changed by the unique characteristics of the sensor device. Expression (16) shows that the optimum number of checkpoints in a cycle is based on three fundamental parameters of the system; $\mathbb{E}[f]$, $\mathbb{E}[P]$, and $\mathbb{E}[D]$. This is consistent with intuition since a decrease in the expected number of failures would mean fewer required checkpoints, an increase in processing time would necessitate more checkpoints, and an increase in the checkpoint overhead would reduce its desirability.

VII. EXPECTATION OF AVERAGE AoI

From (5) we can find an expression for the average AoI of our system by substituting in (13) and (10), and by finding $\mathbb{E}[Y^2]$. Since we have assumed that the variance of each term in Y is known, we are able to use the definition of variance to find $\mathbb{E}[Y^2]$ from

$$\mathbb{E}[Y^2] = \text{Var}(Y) + \mathbb{E}[Y]^2. \qquad (17)$$

By substituting (17) into (5) the expression for the average AoI becomes

$$\mathbb{E}[\Delta] = \frac{\mathrm{Var}(Y)}{2\,\mathbb{E}[Y]} + \frac{3\,\mathbb{E}[Y]}{2} - \mathbb{E}[I]. \qquad (18)$$

As such, by substitution, we find that the average AoI of the system is

$$\mathbb{E}[\Delta]_{\mathrm{MM}} = \frac{\mathrm{Var}(Y)}{2C_3 + \frac{C_4}{\mathbb{E}[h]} + 2D\,\mathbb{E}[h]} - \mathbb{E}[I]$$
$$+ \frac{3}{2}\left(C_3 + \frac{C_4}{2\,\mathbb{E}[h]} + D\,\mathbb{E}[h]\right), \quad (19)$$

where $C_3 = \mathbb{E}[f]C_1 + C_2$ and $C_4 = \mathbb{E}[f]\mathbb{E}[P]$ are defined for compactness of presentation. From (19) we see that the average AoI, as with the average PAoI, is dependent on $\mathbb{E}[f]$ and $\mathbb{E}[h]$. Whilst minimisation of this expression with respect to $\mathbb{E}[h]$ is not presented here, a tractable solution can be found by setting the derivative of (19) to zero and solving for $\mathbb{E}[h]$.

VIII. PEAK AGE OF INFORMATION: MIXED-MEMORY VERSUS SINGLE-MEMORY ARCHITECTURE

Having identified an expression for the average PAoI of our model in (14) we proceed to examine to what extent checkpointing in mixed-memory architecture has affected the performance of our IPD compared to a single-memory device. Whilst in practice, typical commercially available devices are mixed-memory [1, Section 1], we can consider two possible alternatives, a single-memory IPD entirely comprised of NVM [23], [24], or a single-memory IPD entirely comprised of VM. We assume that both types of memory are able to perform processing at the same rate.

Lemma 1. *An entirely NVM IPD will have a lower or equal average PAoI than a checkpointing mixed-memory IPD.*

Proof. Using the same formulation as (7), the *inter-completion time* for such an NVM IPD would be

$$Y_i = I_i + \sum_{j=1}^{f} R_{i,j} + P_i, \qquad (20)$$

since the device does not checkpoint and the only impact of failure is the system off-time. The *completion time* would take the same form as (8). Following the same steps as Sections V and VI, the average PAoI of the device would be

$$\mathbb{E}[\Delta^{\mathrm{Peak}}]_{\mathrm{NVM}} = \mathbb{E}[I] + 2\,\mathbb{E}[f]\,\mathbb{E}[R] + 2\,\mathbb{E}[P]. \qquad (21)$$

We can compare the above expression with the average PAoI of our system by finding the difference between (14) and (21), i.e.

$$\mathbb{E}[\Delta^{\mathrm{Peak}}]_{\mathrm{MM}} - \mathbb{E}[\Delta^{\mathrm{Peak}}]_{\mathrm{NVM}} = 2D\,\mathbb{E}[h]$$
$$+ \mathbb{E}[f]\left(2V + D + 1 + \frac{\mathbb{E}[P]}{\mathbb{E}[h]}\right), \quad (22)$$

which shows that

$$\mathbb{E}[\Delta^{\mathrm{Peak}}]_{\mathrm{NVM}} \le \mathbb{E}[\Delta^{\mathrm{Peak}}]_{\mathrm{MM}} \,\forall\, \mathbb{E}[f], \mathbb{E}[h], \qquad (23)$$

and hence a system comprised of entirely NVM would perform better than or equal to the mixed-memory system. $\square$

Lemma 2. *Under certain environmental conditions a mixed-memory IPD will have a lower average PAoI than a (single-memory) VM IPD.*

Proof. A single-memory device comprised of entirely VM, following the same formulation as (7), would have an *inter-completion time* of

$$Y_i = I_i + \sum_{j=1}^{f}(R_{i,j} + \Gamma_{i,j} + I_{i,j}) + P_i, \qquad (24)$$

where $\Gamma_{i,j}$ is the amount of wasted processing that occurs due to failure j in cycle i and the system does not checkpoint or restore (rather it re-senses after failure). $I_{i,j}$ is the idle time before the system re-senses after failure j. The expected value of $\Gamma_{i,j}$ will be $\mathbb{E}[\Gamma] = \frac{\mathbb{E}[P]+1}{2}$ since the system could waste up to P_i clock ticks of processing per fail and each amount of wasted processing time is equally likely. The *completion time* would take the same form as (8). Following the same steps as Sections V and VI, the average PAoI of a single VM IPD is

$$\mathbb{E}[\Delta^{\mathrm{Peak}}]_{\mathrm{VM}} = 2\,\mathbb{E}[f]\left(\mathbb{E}[R] + \frac{\mathbb{E}[P]+1}{2} + \mathbb{E}[I]\right)$$
$$+ \mathbb{E}[I] + 2E[P]. \quad (25)$$

The above expression can be greater than or smaller than (14) depending on the selected system parameters, hence

$$\exists\, \mathbb{E}[f], \mathbb{E}[h] \ni \mathbb{E}[\Delta^{\mathrm{Peak}}]_{\mathrm{VM}} > \mathbb{E}[\Delta^{\mathrm{Peak}}]_{\mathrm{MM}}, \qquad (26)$$

and thus there is a set of environments in which checkpointing in mixed-memory architecture is more efficient than not checkpointing in single-memory VM architecture. This improvement is most evident when the mixed-memory IPD has a low checkpointing overhead and high failure rate. $\square$

IX. IMPROVING SYSTEM RESILIENCE IN VARIABLE ENVIRONMENTAL CONDITIONS

Thus far we have considered a checkpointing system that can be optimised based on a known expected number of failures, yet in reality IPDs are often placed in environments with variable and unpredictable failure rates—making it difficult to pre-determine an optimum rate of checkpointing. We now propose an alternative method of TDC for mixed-memory devices to improve system resilience—Split-Frequency Checkpointing (SFC)—in which the inter-checkpointing time varies between predefined intervals α and β. Here we once again use the framework devised in Sections II and III. We now assume that the duration of processing between checkpoints, previously $K_{i,n}$, varies between two fixed amounts, $K_{i,\alpha}$ and $K_{i,\beta}$. Then, the processing time $P_i = \sum_{\alpha=1}^{h_\alpha} K_{i,\alpha} + \sum_{\beta=1}^{h_\beta} K_{i,\beta}$. Additionally, the expected wasted processing per fail would be $\mathbb{E}[L] = p_\alpha \mathbb{E}[L_\alpha] + p_\beta \mathbb{E}[L_\beta]$ where p_α and p_β are the probabilities of failure during an α and β checkpoint, respectively, such that $p_\alpha = \frac{\mathbb{E}[K_\alpha]}{\mathbb{E}[K_\alpha]+\mathbb{E}[K_\beta]}$ and $p_\beta = \frac{\mathbb{E}[K_\beta]}{\mathbb{E}[K_\alpha]+\mathbb{E}[K_\beta]}$. $\mathbb{E}[L_\alpha]$ and $\mathbb{E}[L_\beta]$ are the expected wasted processing due to a failure in an α and β checkpoint, respectively, where $\mathbb{E}[L_\alpha] = \frac{\mathbb{E}[K_\alpha]+D+1}{2}$ and $\mathbb{E}[L_\beta] = \frac{\mathbb{E}[K_\beta]+D+1}{2}$. This can also be expressed as

$$\mathbb{E}[L] = \frac{\mathbb{E}[K_{i,\alpha}]^2 + \mathbb{E}[K_{i,\beta}]^2}{2(\mathbb{E}[K_{i,\alpha}] + \mathbb{E}[K_{i,\beta}])} + \frac{D+1}{2}. \qquad (27)$$
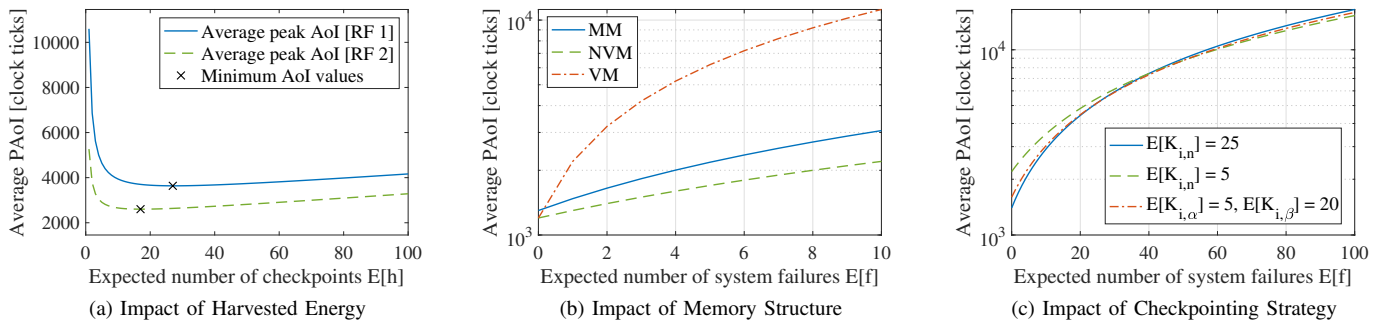
Figure 3. Set of example numerical result (its source code is available at [22]). (a) *Scenario RF 2* conditions require fewer checkpoints than *Scenario RF 1*. Under-checkpointing causes significant increase in average PAoI. (b) Mixed-memory performs better than VM for most failure conditions. (c) SFC is better than inaccurately selected single-frequency.

Following the same derivation of average PAoI as in Sections V and VI, the average PAoI of a SFC system with two frequencies is

$$
\begin{aligned}
E[\Delta^{\text{Peak}}]_{\text{MM(split)}} = {} & C_2 + 2D(\mathbb{E}[h_\alpha] + \mathbb{E}[h_\beta]) \\
& + \mathbb{E}[P] + 2\,\mathbb{E}[f]\left(C_1 + \frac{\mathbb{E}[K_{i,\alpha}]^2 + \mathbb{E}[K_{i,\beta}]^2}{2(\mathbb{E}[K_{i,\alpha}] + \mathbb{E}[K_{i,\beta}])}\right). \quad (28)
\end{aligned}
$$

From this expression we observe that the average PAoI is dependent on a number of system parameters (including $\mathbb{E}[P]$, $\mathbb{E}[f]$, and D), however most interesting is the dependence on inter-checkpointing times $\mathbb{E}[K_{i,\alpha}]$ and $\mathbb{E}[K_{i,\beta}]$, which is notably different to the $\frac{\mathbb{E}[P]}{2\,\mathbb{E}[h]} \to \frac{\mathbb{E}[K]}{2}$ term in (14).

X. NUMERICAL RESULTS

We now provide a set of example numerical results in Fig. 3 using *Scenario RF 1* and *Scenario RF 2* energy harvesting conditions, with data taken from [25, Fig. 1] and summarized in Table I—we note that we have converted ms to our base units of clock ticks herein.

Impact of Harvested Energy. We present the average PAoI of our considered mixed-memory IPD system (expression (14)) in Fig. 3a using the parameters of Table I. From Fig. 3a we see that the average PAoI varies under different energy conditions and that the decrease in $\mathbb{E}[f]$ between *RF 1* and *RF 2* decreases the value of $\mathbb{E}[h]$ that minimises average PAoI. We also see that under-checkpointing has a far more significant impact of data freshness than over-checkpointing.

Impact of Memory Structure. We also consider the relationship between memory architecture and data freshness. Fig. 3b presents plotted expressions (14), (21), and (25) as a function of the expected number of failures $\mathbb{E}[f]$ (using Table I *RF 1* parameters and $\mathbb{E}[h] = 10$ for MM). From Fig. 3b it is evident that, whilst not universally true, for an expected checkpointing overhead and above a low number of failures $\mathbb{E}[\Delta^{\text{Peak}}]_{\text{VM}} > \mathbb{E}[\Delta^{\text{Peak}}]_{\text{MM}} > \mathbb{E}[\Delta^{\text{Peak}}]_{\text{NVM}}$. This shows that whilst entirely NVM architecture will always produce the best possible average PAoI, mixed-memory structures using checkpointing can provide significant improvements in data freshness compared with entirely volatile IPDs.

Impact of Checkpointing Strategy. Finally we show the impact of checkpoining strategy by plotting expressions (14) and (28). Results are presented in Fig. 3c. We see that the system using two inter-checkpointing times ($\mathbb{E}[K_{i,\alpha}] = 5$

Table I
SYSTEM PARAMETER VALUES USED FOR NUMERICAL RESULTS

	$\mathbb{E}[P]^\diamond$	$\mathbb{E}[R]^*$	$\mathbb{E}[f]^*$	$\mathbb{E}[I]^\triangleleft$	$D^\natural$	$V^\dagger$
Scenario RF 1	500	50	15	200	5	10
Scenario RF 2	500	75	6	200	5	10

$\diamond$Set as a baseline for the system. This value varies significantly based on processing needs. *Representative of the dynamic variation of off-time for the first two scenarios in [25, Fig.1] where failure occurs approximately every 50 ms and 100 ms, respectively. $^\triangleleft$Approximate boot time of TinyOS from [25, Section 2]. $^\natural$Overhead can vary significantly in real-world system, 5 ms is of the order of magnitude expected compared with on-time in [25, Fig.1]. †Restoration overhead is typically around twice the checkpoint overhead due to additional management and fixed boot costs.

and $\mathbb{E}[K_{i,\beta}] = 20$) is the most efficient for a range $30 \lessgtr \mathbb{E}[f] \lessgtr 50$ and also provides reasonable performance for all $\mathbb{E}[f]$. Whilst SFC cannot exceed the theoretical optimum for a single frequency (expression (16)) it can provide additional resilience by reducing the risk of a very high average PAoI due to an inappropriately chosen inter-checkpoint interval in an environment with unknown or variable failure rate.

XI. CONCLUSION

In this paper we have considered an Intermittently-Powered Device (IPD) with mixed-memory architecture that periodically checkpoints the system state from volatile memory to non-volatile memory—from which it can be restored should power failure occur. We have identified expressions for the average Age of Information (AoI) and average Peak Age of Information (PAoI) of the system, and found a relationship for the expected checkpointing rate that minimises the expected PAoI. We have also shown that a mixed-memory IPD using Time-Dependent Checkpointing (TDC) can reduce the system PAoI compared with a single volatile memory IPD for selected system parameters. Further, we have proposed an alternative TDC scheme, Split-Frequency Checkpointing, which can improve IPD performance compared with inaccurately selected single-frequency checkpoint intervals.

ACKNOWLEDGEMENT

REFERENCES

[1] B. Lucia, V. Balaji, A. Colin, K. Maeng, and E. Ruppel, "Intermittent computing: Challenges and opportunities," in *Proc. SNAPL*, Asilomar, CA, USA, May 2017.

[2] K. Ganesan, J. San Miguel, and N. Enright Jerger, "The what's next intermittent computing architecture," in *Proc. IEEE HPCA*, Washington, DC, USA, Feb. 2019.

[3] M. Surbatovich, B. Lucia, and L. Jia, "Towards a formal foundation of intermittent computing," in *Proc. ACM OOPSLA*, Chicago, IL, USA, Nov. 2020.

[4] V. Kortbeek, K. S. Yıldırım, A. Bakar, J. Sorber, J. Hester, and P. Pawełczak, "Time-sensitive intermittent computing meets legacy software," in *Proc. ACM ASPLOS*, Lausanne, Switzerland, Mar. 2020.

[5] J. de Winkel, C. Delle Donne, K. S. Yıldırım, P. Pawełczak, and J. Hester, "Reliable timekeeping for intermittent computing," in *Proc. ACM ASPLOS*, Lausanne, Switzerland, Mar. 2020.

[6] J. Hester and J. Sorber, "The future of sensing is batteryless, intermittent, and awesome," in *Proc. ACM SenSys*, Delft, The Netherlands, Nov. 2017.

[7] J. S. Broadhead and P. Pawełczak, "Position paper: Why intermittent computing could unlock low-power visible light communication," in *Proc. Workshop on Light Up the IoT (ACM MobiCom 2020 Workshop)*, London, United Kingdom, Sep. 2020.

[8] S. Kaul, R. D. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *Proc. IEEE INFOCOM*, Orlando, FL, USA, Mar. 2012.

[9] R. D. Yates, Y. Sun, D. R. Brown III, S. K. Kaul, E. Modiano, and S. Ulukus, "Age of information: An introduction and survey," Jul. 2020, arXiv:2007.08564.

[10] C. Kam, S. Kompella, and A. Ephremides, "Age of information under random updates," in *Proc. IEEE ISIT*, Istanbul, Turkey, Jul. 2013.

[11] A. Behrouzi-Far, E. Soljanin, and R. D. Yates, "Data freshness in leader-based replicated storage," in *Proc. IEEE ISIT*, Los Angeles, CA, USA, Jun. 2020.

[12] S. Kaul and R. D. Yates, "Age of information: Updates with priority," in *Proc. IEEE ISIT*, Vail, CO, USA, May 2018.

[13] A. Baknina, S. Ulukus, O. Ozel, J. Yang, and A. Yener, "Sending information through status updates," in *Proc. IEEE ISIT*, Vail, CO, USA, Jun. 2018.

[14] Y. Hsu, E. Modiano, and L. Duan, "Age of information: Design and analysis of optimal scheduling algorithms," in *Proc. IEEE ISIT*, Aachen, Germany, Jun. 2017.

[15] D. Balsamo, A. S. Weddell, A. Das, A. R. Arreola, D. Brunelli, B. M. Al-Hashimi, G. V. Merrett, and L. Benini, "Hibernus++: A self-calibrating and adaptive system for transiently-powered embedded devices," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 35, no. 12, Mar. 2016.

[16] O. Subasi, G. Kestor, and S. Krishnamoorthy, "Toward a general theory of optimal checkpoint placement," in *Proc. IEEE International Conference on Cluster Computing*, Honolulu, HI, USA, Sep. 2017.

[17] O. Ozel, "Timely status updating through intermittent sensing and transmission," in *Proc. IEEE ISIT*, Los Angeles, CA, USA, Jun. 2020.

[18] S. Jayasekara, A. Harwood, and S. Karunasekera, "A utilization model for optimization of checkpoint intervals in distributed stream processing systems," *Future Generation Computer Systems*, vol. 110, Apr. 2020.

[19] S. Di, M. S. Bouguerra, L. Bautista-Gomez, and F. Cappello, "Optimization of multi-level checkpoint model for large scale HPC applications," in *Proc. IEEE Int. Parallel and Distributed Processing Symp.*, Phoenix, AZ, USA, Aug. 2014.

[20] P. Rafiee and O. Ozel, "Active status update packet drop control in an energy harvesting node," in *Proc. IEEE SPAWC*, Atlanta, GA, USA, May 2020.

[21] S. I. Resnick, *Adventures in Stochastic Processes*. Birkhäuser, 1992.

[22] J. S. Broadhead and P. Pawełczak, "Source code for numerical results of this paper," https://github.com/tudssl/intermittency-aoi, Feb. 2021, last accessed: May. 8, 2021.

[23] F. Su, K. Ma, X. Li, T. Wu, Y. Liu, and V. Narayanan, "Nonvolatile processors: Why is it trending?" in *Proc. ACM/IEEE DATE*, Lausanne, Switzerland, Mar. 2017.

[24] T.-K. Chien, L. Chiou, C.-C. Lee, Y.-C. Chuang, S.-H. Ke, S.-S. Sheu, H.-Y. Li, P.-H. Wang, T.-K. Ku, M.-J. Tsai, and C.-I. Wu, "An energy-efficient nonvolatile microprocessor considering software-hardware interaction for energy harvesting applications," in *Proc. IEEE VLSI-DAT*, Hsinchu, Taiwan, Apr. 2016.

[25] B. Ransford, J. Sorber, and K. Fu, "Mementos: System support for long-running computation on RFID-scale devices," in *Proc. ACM ASPLOS*, Newport Beach, CA, USA, Mar. 2011.