Real-Time Reconstruction of a Counting Process Through First-Come-First-Serve Queue Systems

Meng Wang, *Student Member, IEEE*, Wei Chen, *Senior Member, IEEE*, and Anthony Ephremides, *Life Fellow, IEEE*

***Abstract*— For the emerging Internet of Things (IoT), one of the most critical problems is the real-time reconstruction of signals from a set of aged measurements. During the reconstruction, distortion occurs between the observed signal and the reconstructed signal due to sampling and queuing delay. We focus on minimizing the average distortion defined as the 1-$norm$ of the difference of the two signals under the scenario that a Poisson counting process is reconstructed in real-time on a remote monitor. We consider the reconstruction under three special sampling policies. For each of the policy, we derive the closed-form expression of the average distortion by dividing the overall distortion area into polygons and analyzing their structures. It turns out that the polygons are built up by sub-polygons that account for distortions caused by sampling and queuing delay. The closed-form expressions of the average distortion help us find the optimal sampling parameters that achieve the minimum distortion. In addition, we propose an interpolation algorithm to further decrease the average distortion and give its lower-bound on distortion for one of the three sampling policies. Simulation results are provided to validate our conclusion.**

***Index Terms*— Internet of Things, real-time signal reconstruction, average distortion, age of information, sampling, queuing delay, Poisson process, Markov chain, $D/M/1$, $E_r/M/1$.**

I. Introduction

RECENTLY, the new emerging Internet of Things (IoT) has gained massive attraction in various spheres. It is common to reconstruct on-going signals (or functions of the signals) remotely in the IoT scenarios [1], [2]. However, the reconstruction is usually based on the measurements that are under sampled and delayed in the network [3]. The under sampled measurements omit the detailed information of the original process and the delayed measurements against the requirement of real-time. Thus, it is an important and interesting work to study how to improve the performance of the reconstruction from a set of under-sampled and aged measurements. Solving this problem brings more challenges to the traditional sampling and signal processing techniques.

Specifically, on one hand, adopting the Nyquist rate when sampling is undesirable as it will cause crowded traffic in the network and/or sometimes impossible when the signal of interest is analog and without a clear beginning and end [4]. On the other hand, in traditional non-causal reconstruction scheme, one needs the signal's samples from the entire time horizon, both in the past and in the future. Perfect reconstruction is achievable but with significant delay spent on waiting for future samples to be taken and delivered [5], [6]. To meet the requirement of real-time, it is more suitable to adopt the causal reconstruction scheme, where one can use the available past samples to estimate and predict the current signal value and the goal is to minimize the reconstruction error. Due to the above mentioned reasons, it is of great importance to study the real-time signal reconstruction with sub-Nyquist rate.

Many works have been done in this area. On one hand, some of them are interested in the compressive sampling (CS) which is a recently evolved notion [7]–[9]. By recognizing that the intrinsic information of an analog signal is not solely dictated by its bandwidth, CS provides ways to sample a large class of signals at rates which are far below the Nyquist rate and then reconstruct them asymptotically and exactly via efficient numerical methods [10]–[12]. In [13] and [14], the authors proposed a suite of dynamic updating algorithms for solving the ℓ_1-norm minimization problems for the case where the central processing unit receives a continuous stream of measurements (or samples) acquired at a fixed rate. In [15], reconstruction algorithm based on sliding window processing was proposed to improve the performance of the signal recovery. The proposed methods in [13]–[15] provide state-of-the-art performance for the case where central processing unit receives a continuous stream of samples acquired at a fixed rate. However, a standard CS approach often assumes that the signal of interest is a vector of finite length (i.e., already

Manuscript received January 22, 2019; revised July 23, 2019; accepted January 5, 2020. Date of publication February 24, 2020; date of current version June 18, 2020. Meng Wang and Wei Chen were supported in part by the National Science Foundation of China under Grant 61971264, in part by the Beijing Natural Science Foundation under Grant 4191001, and in part by the National Program for Special Support for Eminent Professionals of China (10000-talent program). Anthony Ephremides was supported in part by the U.S. Office of Naval Research under Grant N000141812046 and in part by the U.S. National Science Foundation under Grant CCF1420651, Grant CNS1526309, and Grant CCFR1813078. This paper was presented in part at the 2019 IEEE International Conference on Communications (ICC). *(Corresponding author: Wei Chen.)*

Meng Wang and Wei Chen are with the Department of Electrical Engineering, Tsinghua University, Beijing 100084, China, and also with the Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: mengwangee@gmail.com; wchen@tsinghua.edu.cn).

Anthony Ephremides is with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20740 USA, and also with the Institute for System Research, University of Maryland, College Park, MD 20740 USA (e-mail: etony@umd.edu).

Communicated by L. Ying, Associate Editor for Communication Networks.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2020.2976013

digitized) and reconstructs it based on the previously acquired samples (i.e., the non-causality assumption in classic sampling theory [8]). Thus, methods that release these constraints need to be further explored.

On the other hand, some works have been done by focusing on deriving the age-aware sampling and reconstruction techniques [16], [17]. The age of information (AoI) is a topic first proposed in [18] and further studied in [19]–[21] to describe the freshness of the information. To perform causal reconstruction, one commonly used method is to introduce a delay in the reconstruction signal and truncate the remaining non-causal part. Thus, the age of the received samples has an impact on the reconstruction quality of the signal. In [16] and [17], the authors minimized the reconstruction error by solving the instances of the causal sampling problem, and provided better performance than the classic uniform sampling approach. In these works, the defined AoI is a process-independent metric. However, the freshness of the information should depend on the context, including the correlation structure, regular pattern, and some parameter values of the signal, which varies from one process to another. Thus, the original defined AoI needs modification to better measure the freshness of information and play an important role in signal reconstruction.

We focus on the remote and real-time process reconstruction problem with a new defined age-related metric, i.e., the average distortion, being the measurement criteria. To understand and make progress in the difficult issue of studying the age-related component of distortion (which is new and unstudied so far), the Poisson counting process is a good first choice since it is relatively simple, tractable, and displays all the aspects of distortion that age induces. Thus, we consider such a system that measurements of an on-going Poisson counting process are sampled and transmitted to a remote monitor, where a new process is reconstructed based on the received aged measurements as an estimation of the original process. We adopt the average distortion defined as the average distance between the original process and the reconstructed process as the metric to evaluate the reconstruction error. In this case, the distortion happens due to sampling and queuing delay. On one hand, sampled measurements ignore the evolution of the process of interest during the sampling interval and just provide snapshots at the sampling instants. On the other hand, the measurements cannot be received by the remote receiver immediately after being generated due to the queuing behavior which causes inevitable delay. These two kinds of distortion happen simultaneously, twist with each other, and build up the total reconstruction distortion. From the perspective of minimizing the average distortion, we need to jointly optimize the sampling policy and reconstruction policy. However, this brings much challenge to our study, since we are exploring the un-studied area of process-dependent AoI. As a start, we focus on minimizing the average distortion under a fixed reconstruction policy and three special sampling policies. And We want to find out what are the optimal sampling parameters that achieve the minimum average distortion. To this end, the closed-form expressions of the average distortion are built as functions of the sampling parameters. Simulation results validate the correctness of these distortion functions. The contributions of this work are summarized as follows.

1) The Definition of the Average Distortion: The defined average distortion covers the distortions caused by sampling and queueing delay, that occur simultaneously and twist with each other during the real-time reconstruction. At the mean time, it is a process-dependent metric which allows it to better measure the freshness of the received samples compared to the AoI metric.

2) The Closed-Form Expressions of the Average Distortion Under Different Sampling Policies: We consider three special sampling policies, namely, the uniform policy which samples the original process periodically, the threshold-based policy which performs one sampling only when the update of original process accumulates to a certain positive threshold, and the zero-wait policy which samples only when the queue server is idle. For each policy, we derive the closed-form expression of the average distortion. Especially, for the uniform policy and threshold-based policy, sampling with a smaller period or threshold will cause greater queueing delay but smaller sampling distortion, while adopting a greater period or threshold will cause smaller queueing delay but greater sampling distortion. The closed-form distortion expressions help us find the optimal sampling parameters that balance this tradeoff.

3) The Interpolation Algorithm: An interpolation algorithm is provided to further decrease the average distortion by inserting estimations for the omitted sampling details. The lower-bound on distortion of the algorithm is obtained for uniform sampling policy.

The paper is organized as follows. Section II describes the system model and gives the definition of the average distortion. Section III gives the closed-form expressions of the average distortion under three special sampling policies. Section IV proposes an interpolation algorithm to further decrease the average distortion and gives the lower-bound for uniform policy. Simulation results are given in Section V to validate the theoretical results. Finally, Section VI concludes this work and talks about the future work.

II. System Model

We consider the scenario that a counting process is reconstructed on a remote receiver in real-time as shown in Fig. 1. The original counting process is a Poisson process and denoted by $N(t)$. The sampler measures $N(t)$ and generates sample packets. These sample packets have to wait in a buffer queue before being sent out. At the remote monitor, a process $\hat{N}(t)$ is reconstructed as an approximation of $N(t)$ based on the received samples.

We plot the original process $N(t)$ in Fig. 2(a), the uniform sampling in Fig. 2(b), and the reconstructed process $\hat{N}(t)$ in Fig. 2(c). We then introduce the specific system parameters with the aid of these schematic plots.

Assume that the average arrival rate of Poisson process $N(t)$ is λ. In this paper, we regard one counting increase of $N(t)$ is an event arrival. Let $s_0 = 0$ and s_n denote the arrival time of the nth event where n is a positive integer. Thus, there is $s_{n-1} \leqslant s_n, \forall\, n \geqslant 1$. Let $\{X_n, n \geqslant 1\}$ denote the sequence of interarrival times, namely, $X_n = s_n - s_{n-1}, \forall\, n \geqslant 1$. Thus,

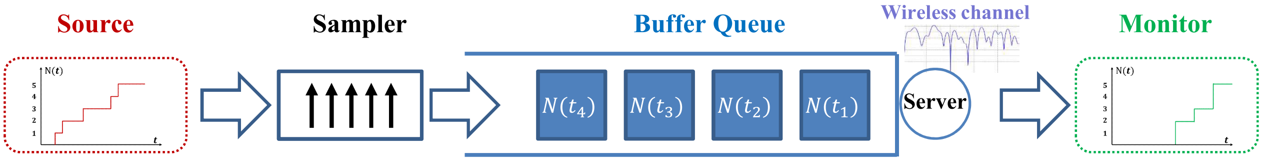
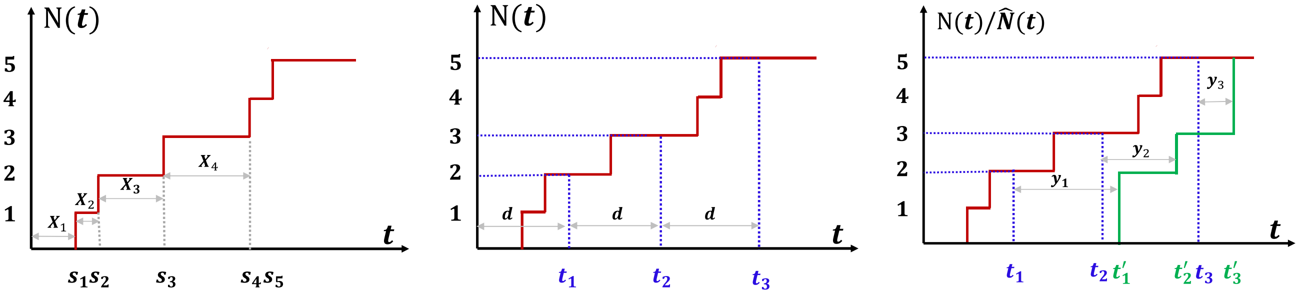


Fig. 1. System model.



(a) The oringinal counting process (b) Uniform sampling (c) The reconstructed process

Fig. 2. The illustration of sampling and reconstruction.

the interarrival times X_n $(n \geqslant 1)$ are independent identically distributed $(i.i.d.)$ exponential random variables with mean $\frac{1}{\lambda}$.

Let $\{t_i, i \geqslant 1\}$ denote the sampling time sequence. The samples measures $N(t)$ and generates a sample packet[1] denoted by $N(t_i)$ at t_i $(i \geqslant 1)$. We consider three sampling polices, namely, the uniform policy which samples $N(t)$ periodically, the threshold-based policy which performs one sampling only when the update of $N(t)$ accumulates to a positive integer threshold β, and the zero-wait policy which samples only when the queue server is idle. Suppose that the uniform sampler works with d $(d > 0)$ being the sampling interval. The sampling rate r is defined as $r = \frac{1}{d}$. Thus, the sampling time sequences for uniform sampling are $\{t_i = id, i \geqslant 1\}$. We regard the threshold-based policy and the zero-wait policy as non-uniform policies.

We assume that the service time for each sample packet is an $i.i.d.$ exponential random variable with mean $\frac{1}{\mu}$, which accounts for the wireless transmission time of each packet. The probability dense function $(p.d.f)$ of the service time is then given as

$$f(x) = \mu e^{-\mu x}, \ x \geqslant 0. \tag{1}$$

After being generated at time t_i, the ith sample packet enters the queue and waits for its turn to be served. The waiting delay w_i is defined as the waiting time of the ith packet. At time $(t_i + w_i)$, the server starts to serve this packet and the service time is denoted by v_i. Suppose the ith packet finishes its serve and arrives at the monitor at $t^{'}_i$, the queueing delay is defined as the sum of waiting delay and service time, namely, $(w_i + v_i)$, which also equals to $(t^{'}_i - t_i)$. We assume the first-in-first-out queueing rule, i.e., $t^{'}_i < t^{'}_{i+1}$. At the monitor, the reconstruction rule is that $\hat{N}(t)$ remains the value of last received sample and updates to a new value with a newly arrived sample, i.e.,

$$\hat{N}(t) = \begin{cases} N(t_{i-1}), & t^{'}_{i-1} < t < t^{'}_i; \\ N(t_i), & t^{'}_i \leqslant t < t^{'}_{i+1}, \end{cases} \tag{2}$$

where $t^{'}_{i-1}$, $t^{'}_i$ and $t^{'}_{i+1}$ denote the arrival time of the $(i{-}1)$th, ith and $(i{+}1)$th sample packets. As shown in Fig. 2(c), distortion occurs between the original process $N(t)$ and the reconstructed process $\hat{N}(t)$.

Definition 1: Let $\hat{N}(t) = 0, t < t_1$. The real-time distortion $D(t)$ is defined as the difference between $N(t)$ and $\hat{N}(t)$, i.e.,

$$D(t) = N(t) - \hat{N}(t), \ t \geqslant 0. \tag{3}$$

Considering that $N(t)$ is a counting process and there exists non-negative delay between $N(t)$ and $\hat{N}(t)$, we conclude that $D(t) \geqslant 0, \ \forall \ t \geqslant 0$. The real-time distortion occurs for two reasons. On one hand, instead of recording the arrival time of every event of $N(t)$, the sampling omits the details of $N(t)$ during every sampling interval and simply provides the result of $N(t)$ at the sampling instant. On the other hand, the sampled measurements cannot be received by the monitor without delay which occurs due to the waiting time and the service time. Suppose that T is a great positive real number, we consider the reconstruction during time horizon $[0, T]$.

Definition 2: The average distortion is defined as the 1-$norm$ of function $D(t)$, namely,

$$\Theta = \frac{1}{T}\int_0^T D(t)\mathrm{d}t. \tag{4}$$

From *Definition* 2, the average distortion represents the average gap between the two curves in Fig. 2(c). We evaluate the performance of the reconstruction with the average distortion

[1] Suppose that the increase of $N(t)$ during one sampling interval can be exactly described by one packet. This assumption is ideal since the length of one packet is limited and the arrival of the original process during one sampling interval is possible infinite. More detailed discussion can be found in *Appendix* A.

TABLE I
THE AVERAGE DISTORTION UNDER THREE SAMPLING POLICIES

Policy		Parameter	Average distortion
uniform sampling policy		sampling rate r	$\Theta(r)=\lambda\left(\frac{1}{2r}+\frac{\sigma}{\mu(1-\sigma)}+\frac{1}{\mu}\right)$
non-uniform	threshold-based policy	sampling threshold β	$\Theta(\beta)=\lambda\left(\frac{\beta-1}{2\lambda}+\frac{1}{\mu(z_0^{\beta}-1)}+\frac{1}{\mu}\right)$
	zero-wait policy	\	$\Theta_{zw}=\lambda\frac{2}{\mu}$

TABLE II
NOTATIONS

Symbol	Description	Symbol	Description
$N(t)$	the original process	$\hat{N}(t)$	the reconstructed process
s_n	the arrival time of nth event	$\{X_n\}$	the interarrival time sequence
λ	the arrival rate of $N(t)$	μ	the service rate of queue server
t_i	the sampling time of ith sample	$t_i^{'}$	the arriving time of ith sample
d	the sampling period	r	the sampling rate
β	the sampling threshold	$f(x), p(x), h(x)$	$p.d.f.$s of random variables
w_i	the waiting time	v_i	the service time
$D(t)$	the reconstruction distortion	$\Theta, \breve{\Theta}, \Theta(\cdot), \Theta_{(\cdot)}$	the average distortion
σ	the solution of a Lambert W function	z_0	one of the zeroes of a polynomial
$S_{(\cdot)}$ or $\hat{S}_{(\cdot)}$	the area of a polygon	$\mathbb{E}\{\cdot\}$	the expectation operator

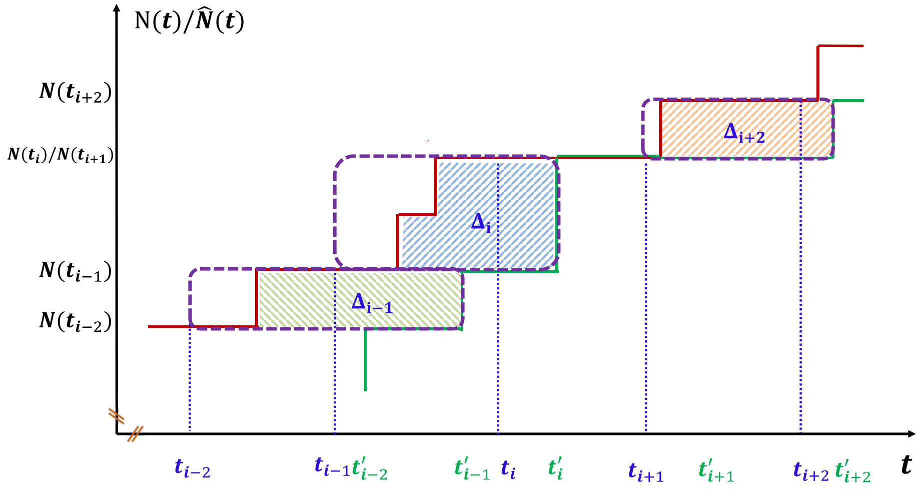


Fig. 3. The divided polygons under uniform sampling policy: the purple dash rectangles are the borders of the defined sets $\{R_i,\ i \geqslant 1\}$. The shaded areas are the divided polygons. Notice that, the area of one polygon can be zero if no event happen during corresponding sample interval. Because in this case, $N(t_{i-1}) = N(t_i)$ which leads to the rectangle area collapse to a horizontal line, such as Δ_{i+1} in the figure.

in (4) being the criterion and try to minimize it by finding the optimal sampling parameters for different sampling policies.

As shown in TABLE I, the key parameter is the sampling rate r (also the sampling period d, since $d = \frac{1}{r}$) for uniform policy and the sampling threshold β for threshold-based policy. Thus, we build the average distortion as functions of these key parameters. The zero-wait policy is a fix policy, we provide the average distortion functions for reference. In the following section, we give the detailed derivation of the conclusion in TABLE I. Some notations are given in TABLE II for convenience.

III. THE AVERAGE DISTORTION FUNCTIONS

We derive the closed-form expressions of the average distortion under the uniform policy, the threshold-based policy, and the zero-wait policy in three subsections, respectively.

A. The Average Distortion-Sampling Rate Function

For uniform sampling policy, the average distortion Θ is merely decided by the sampling rate r for a given original process. Thus, we formulate the average distortion-sampling rate function $\Theta(r)$, with which the optimal sampling rate can be obtained to minimize the average distortion.

As shown in Fig. 3, the overall distortion area between the two curves can be divided into $I = \frac{T}{d}$ polygons[2] which are marked as $\{\Delta_i, 1 \leqslant i \leqslant I\}$ as the shaded areas. Let (x, y) denote a point in the plane. Set R_i defined as $\{(x, y) \mid t_{i-1} \leqslant x < t_i^{'},\ N(t_{i-1}) \leqslant y < N(t_i)\}$ represents a rectangle area in the plane. Polygon Δ_i is the distortion area enclosed by rectangle area R_i. By reducing index i in the definition of set R_i to $(i-1)$, we can obtain set R_{i-1} and polygon Δ_{i-1} in the similar way. By increasing index i to $(i+1)$, we can

[2] Suppose that T can be exactly divided by d.

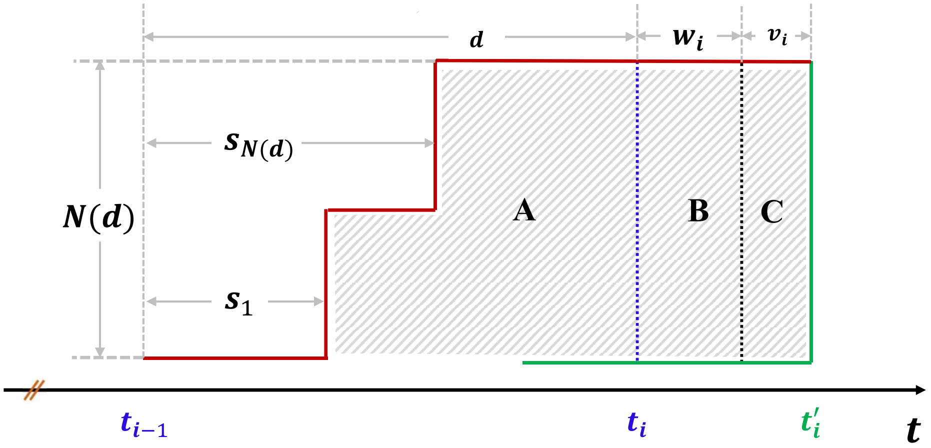


Fig. 4. Polygon Δ_i: d is the average sampling interval, $N(d)$ is the height of the polygon (i.e., the number of events occurring in sampling interval d), w_i is the waiting time of sample i in the queue and v_i is the service time of the ith sample i.

obtain polygon Δ_{i+1}. With the above definition, the divided polygons will not overlap and all the polygons build up the total distortion area.

Lemma 1: The average distortion-sampling rate function under uniform sampling is given by

$$\Theta(r) = r\mathbb{E}\{S_\Delta\}, \tag{5}$$

where S_{Δ_i} is the area of polygon Δ_i and $\mathbb{E}\{S_\Delta\}$ is the average of S_{Δ_i}.

Proof: Based on the definition in (4), the average distortion is the average gap between the two curves. With the division in Fig. 3, it can be formulated as

$$\Theta(r) = \frac{1}{T}\int_0^T D(t)\mathrm{d}t = \frac{I}{T}\frac{1}{I}\sum_{i=1}^{I} S_{\Delta_i}. \tag{6}$$

In (6), the item $\frac{I}{T}$ is sampling rate r and the item $\frac{1}{I}\sum_{i=1}^{I} S_{\Delta_i}$ is the average area of the divided polygons. Thus, we arrive at the average distortion in (5). ■

Lemma 2: The average area of the divided polygons is given as

$$\mathbb{E}\left\{S_\Delta\right\} = \frac{\lambda}{2r^2} + \frac{\lambda}{r}\left(\frac{\sigma}{\mu(1-\sigma)} + \frac{1}{\mu}\right), \tag{7}$$

where parameter σ is the solution of Lambert W Function, namely,

$$\sigma = -\frac{r}{\mu}\mathcal{W}(-\frac{\mu}{r}e^{-\frac{\mu}{r}}). \tag{8}$$

Proof: As shown in Fig. 4, by analyzing the structure of the polygon, we divide it into three sub-polygons marked as A, B, and C based on time (horizontal axis) that correspond to the distortions caused by sampling, waiting time, and service time. Specifically, starting from t_{i-1} and ending at t_i, sub-polygon A is the distortion caused by sampling; starting from t_i and ending at $(t_i + w_i)$, sub-polygon B is the distortion caused by waiting delay; and starting at $(t_i + w_i)$ and ending at $(t_i + w_i + v_i)$, sub-polygon C is the distortion caused by the service time. Thus, we have

$$\mathbb{E}\{S_\Delta\} = \mathbb{E}\{S_A\} + \mathbb{E}\{S_B\} + \mathbb{E}\{S_C\}, \tag{9}$$

where S_A, S_B, and S_C represents the area of sub-polygons A, B, and C, respectively. It means we are able to calculate $\mathbb{E}\{S_\Delta\}$ as the sum of the average areas of the three divided sub-polygons, which are given in *Appendix* B for brevity. Then, we arrive at (7). ■

Theorem 1: The average distortion-sampling rate function is given as

$$\Theta(r) = \lambda\Big(\frac{1}{2r} + \frac{\sigma}{\mu(1-\sigma)} + \frac{1}{\mu}\Big). \tag{10}$$

Proof: Combining the results in *Lemma* 1 and *Lemma* 2, we arrive at the conclusion. ■

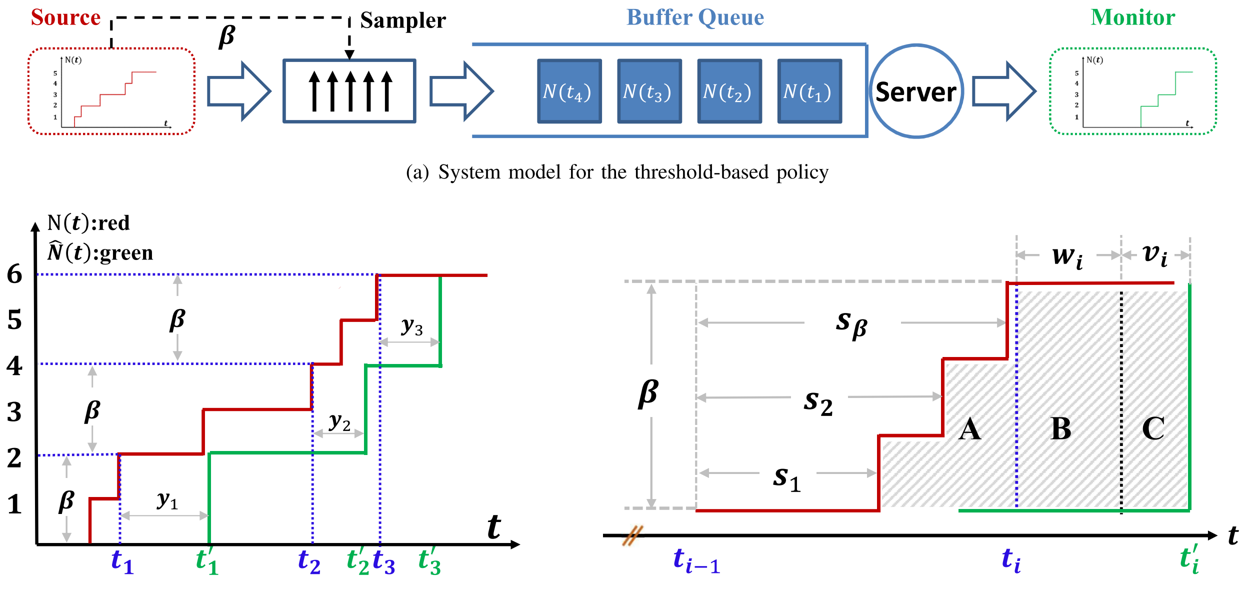
In *Theorem* 1, we give the average distortion-sampling rate function. The optimal sampling rate is the one that gives the minimum average distortion, i.e.,

$$r^* = \arg\min_r \quad \Theta(r). \tag{11}$$

Since $\Theta(r)$ only has one variable, i.e., the sampling rate r, one can find the optimal sampling rate r^* by calculating its derivation and making it as zero. However, it is not explicit to obtain the derivation due to the existing of parameter σ. In the simulation section, we can plot the curve of function $\Theta(r)$ and choose a sampling rate that induces a smaller average distortion.

We consider the optimal sampling rate from another point of view. As we have illustrated that the distortion occurs due to sampling and queueing delay. These two kinds of distortion occur simultaneously and twist with each other.[3] Thus, there exists an interesting tradeoff between these two kinds of distortion when we adjust the sampling rate. Specifically, if we sample with a smaller r, the distortion caused by sampling will increase since the description of $N(t)$ is less elaborate, while the distortion caused by transmission will decrease since less packets will arrive at the queue. On the contrary, if we sample with a greater r, the distortion cased by sampling will decrease

[3] One can imagine that, for the reconstructed process $\hat{N}(t)$, it only suffers from shape change if the reconstruction is done locally or only suffers from time translation if the sampler always keeps an eye on $N(t)$ and updates for every count.



(a) System model for the threshold-based policy

(b) Average distortion

(c) Divided polygon

Fig. 5. The system model, the overall distortion, and the divided polygons under the threshold-based policy.

while the distortion caused by transmission will increase since more arrival packets lead to higher waiting time. The optimal sampling rate r^* is the rate that balances this tradeoff.

B. The Average Distortion-Sampling Threshold Function

We then focus on minimizing the average reconstruction distortion under threshold-based policy. As shown in Fig. 5(a), the sampler only performs one sampling when the accumulation of the original process $N(t)$ equals the given threshold β ($\beta > 0$). The threshold β is the only key parameter of the threshold-based sampling, since it determines the frequency of sampling and the traffic load of the queueing system, which are two critical factors of the reconstruction distortion. We derive the closed-form expression of the average distortion-sampling threshold function $\Theta(\beta)$.

In Fig. 5(b), we give the curves of the original process $N(t)$ and the reconstructed process $\hat{N}(t)$. The gap between $N(t)$ and $\hat{N}(t)$ is the overall reconstruction distortion. It can be divided into I polygons, where I is the total number of sampling during the horizon $[0, T]$. Each polygon is denoted by Δ_i and defined in the similar way as we introduce in Section III-A. By defining the rectangle set $R_i = \{(x, y) \mid t_{i-1} \leqslant x < t_i^{'}, N(t_{i-1}) \leqslant y < N(t_i)\}$ and using it to overlap the overall distortion area, we obtain the polygon Δ_i and plot it in Fig. 5(c).

Lemma 3: The average distortion under the threshold-based sampling policy is reformulated as

$$\Theta(\beta) = \frac{\lambda}{\beta}\mathbb{E}\left\{S_\Delta\right\}, \tag{12}$$

where S_{Δ_i} is the area of polygon Δ_i and $\mathbb{E}\left\{S_\Delta\right\}$ is the average of S_{Δ_i}.

Proof: Based on (4), the average distortion $\Theta(\beta)$ can be rewritten as

$$\Theta(\beta) = \frac{1}{T}\int_0^T D(t)\mathrm{d}t = \frac{I}{T}\frac{1}{I}\sum_{i=1}^{I} S_{\Delta_i}. \tag{13}$$

In (13), the term $\frac{I}{T}$ can be rewritten as $\frac{\lambda}{\beta}$ since samples are obtained for every β events and the term $\frac{1}{I}\sum\limits_{i=1}^{I} S_{\Delta_i}$ is the average area of the divided polygons. ■

Lemma 4: The average area of the polygons is given as

$$\mathbb{E}\left\{S_\Delta\right\} = \beta\left(\frac{\beta-1}{2\lambda} + \frac{1}{\mu(z_0^\beta - 1)} + \frac{1}{\mu}\right), \tag{14}$$

where z_0 is one of the zeroes of polynomial $\frac{\lambda}{\mu}z^{\beta+1} - (1 + \frac{\lambda}{\mu})z^\beta + 1$, whose modulus is greater than one,[4] i.e., $|z_0| > 1$.

Proof: As shown in Fig. 5(c), a polygon can be divided into three sub-polygons denoted by A, B, and C, that account for the sampling distortion, the distortion caused by waiting time, and the distortion caused by service time. Thus, the average of S_Δ can be calculated as $\mathbb{E}\left\{S_\Delta\right\} = \mathbb{E}\left\{S_A\right\} + \mathbb{E}\left\{S_B\right\} + \mathbb{E}\left\{S_C\right\}$. We calculate that $\mathbb{E}\left\{S_A\right\} = \frac{(\beta-1)\beta}{2\lambda}$ and $\mathbb{E}\left\{S_B\right\} = \frac{\beta}{\mu(z_0^\beta - 1)}$, whose derivations are given in *Appendix* E for brevity. And we have $\mathbb{E}\left\{S_C\right\} = \beta\mathbb{E}\left\{v_i\right\} = \frac{\beta}{\mu}$. Summing these three items up, we arrive at the conclusion in (14). ■

Theorem 2: The average distortion-sampling threshod function is give as

$$\Theta(\beta) = \frac{\lambda}{\beta}\mathbb{E}\left\{S_\Delta\right\} = \lambda\left(\frac{\beta-1}{2\lambda} + \frac{1}{\mu(z_0^\beta - 1)} + \frac{1}{\mu}\right). \tag{15}$$

[4] There is only one zero whose modulus is greater than one, namely, z_0, there is one zero whose modulus is equal one, and modulus of the left $\beta - 1$ zeroes are less than one.

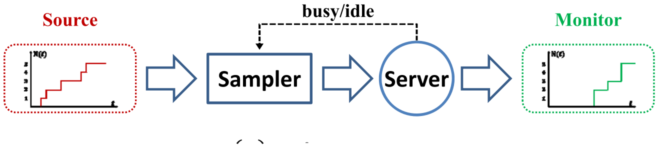


Fig. 6. The system model with zero-wait sampling policy.

Proof: Combining the results in (12) and (14), we arrive at the conclusion. ■

The optimal threshold is the the one that leads to the minimum average distortion, i.e.,

$$\beta^* = \arg\min_{\beta} \ \Theta(\beta). \tag{16}$$

Similar to the relationship between the average distortion and the sampling rate for uniform sampling policy, there exists similar relationship between the average distortion and the sampling threshold. That is, a greater threshold will evidently decrease the traffic load of the queueing system, as well as the queueing delay experienced by the samples. Thus, the distortion caused by queueing will be decreased. At the mean time, this greater threshold will increase the distortion caused by sampling. There is similar result for a smaller threshold. Thus, the optimal sampling threshold β^* is the one that balances the tradeoff between the distortions caused by sampling and transmission.

C. The Average Distortion Under Zero-Wait Sampling Policy

In this subsection, we focus on the reconstruction with zero-wait sampling policy. For zero-wait sampling policy, the sampler only measures the original process when the queue server is idle, as shown in Fig. 6. Thus, the sampled packets will be served directly once it is generated, instead of waiting in the queue compared to the cases in uniform sampling and in threshold-based policy.

In Fig. 7(a), we plot the original process $N(t)$, the reconstructed process $\hat{N}(t)$, as well as the sampling operation. One can see that, the sampling time of the ith packet t_i overlaps with the service completion time of the $(i-1)$th packet $t^{'}_{i-1}$. The overall distortion area between the two curves can be divided into polygons, one of which is denoted by Δ_i and given in Fig. 7(b).[5] Polygon Δ_i is the overlap area between the overall distortion and rectangle area defined by $R_i = \{(x, y) \mid t_{i-1} \leqslant x < t^{'}_i,\ N(t_{i-1}) \leqslant y < N(t_i)\}$.

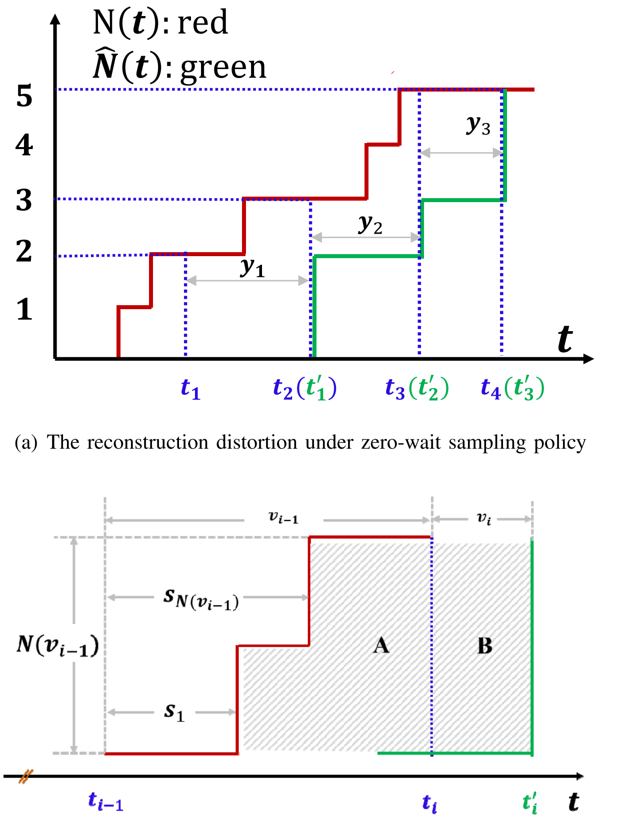
Lemma 5: The average distortion under zero-wait sampling policy is denoted by Θ_{zw} and can be calculated as

$$\Theta_{zw} = \mu \mathbb{E}\left\{S_\Delta\right\}, \tag{17}$$

where $\mathbb{E}\left\{S_\Delta\right\}$ is the average area of the divided polygons.

Proof: Based on (4), Θ_{zw} can be reformulated as

$$\Theta_{zw} = \frac{1}{T}\int_0^T D(t)\mathrm{d}t = \frac{I}{T}\frac{1}{I}\sum_{i=1}^{I}\Delta_i. \tag{18}$$



(a) The reconstruction distortion under zero-wait sampling policy

(b) Polygon Δ_i: it can be divided into two sub-polygons.

Fig. 7. The sampling and reconstruction under zero-wait sampling policy.

Recall the division of the polygons, the total number of the polygons I equals the total sampling times as well as the total transmission times. Thus, the first item in (18) equals μ. The term $\sum_{i=1}^{I}\Delta_i$ is the average area of the polygons. Thus, we arrive at (17). ■

Lemma 6: The average area of the polygons is given as

$$\mathbb{E}\left\{S_\Delta\right\} = \frac{2\lambda}{\mu^2}. \tag{19}$$

Proof: As shown in Fig. 7(b), a polygon Δ_i can be divided int sub-polygons marked as A and B, that account for the sampling distortion and the distortion caused by service time, respectively. Thus, we have $\mathbb{E}\left\{S_\Delta\right\} = \mathbb{E}\left\{S_A\right\} + \mathbb{E}\left\{S_B\right\}$. In *Appendix* F, we show that $\mathbb{E}\left\{S_A\right\} = \mathbb{E}\left\{S_B\right\} = \frac{\lambda}{\mu^2}$. Thus, we obtain (19). ■

Theorem 3: The average distortion under zero-wait policy is given as

$$\Theta_{zw} = \lambda\frac{2}{\mu}. \tag{20}$$

Proof: Combining the results in (17) and (19), we obtain the result in (20). ■

Since we are calculating the average distortion for a fixed sampling policy, i.e., zero-wait sampling policy, the average distortion is a constant value, given the arrival rate of the original process λ and the server capacity μ. Zero-wait sampling

[5] Since sample $N(t_i)$ is obtained at t_i immediately after the previous sample is served, the sampling interval $(t_{i+1} - t_i)$, the service time of sample $N(t_i)$, i.e., v_i in Fig. 7(b), and the inter-arrival time of samples $N(t_i)$ and $N(t_{i+1})$, i.e., y_i in Fig. 7(a), are exactly same and can be captured by $i.i.d.$ exponential random variables, whose $p.d.f$ are given in (1).

policy is simple and easy to apply since there is no need to consider the buffer capacity of the server which is quite limit for portable devices.

IV. Reconstruction With Interpolation Algorithm

We then propose an algorithm that can further decrease the average distortion based on interpolation. The lower bound of the average distortion is provided as the upper bound of the performance of these interpolation algorithms for uniform sampling policy.

A. Interpolation Algorithms

The interpolation algorithms are based on the following idea: although the sampling omits the details of the original process during the sampling interval, the monitor still can make some estimations for the omitted details, i.e., reconstruction with interpolation.

By interpolation, we mean that, if the monitor receives one packet at time $t_i^{'}$ which indicates any increase compared to the last received packet at time $t_{i-1}^{'}$, the monitor can insert some updates itself during interval $(t_{i-1}^{'}, t_i^{'})$ to decrease the reconstruction distortion. An example of this idea is given in Fig. 8, where the new received packet at time $t_i^{'}$ indicates that the original process is increased by 2 compared to the last sample, but we don't know exactly when do these events occur. In this case, estimation can be made for the occurring time of these events. For example, at time $t_{i_1}^{'}$, the reconstructed process $\hat{N}(t_{i_1}^{'})$ is set to $N(t_{i-1}^{'})+1$. As a result, the overall distortion is decreased by the area labeled with orange color. Taking uniform sampling policy for example, we give an algorithm in *Algorithm* 1 that adopts uniform guessing for the omitted details, which means that the inserted points are chosen to be uniformly distributed on the open interval $(t_{i-1}^{'}, t_i^{'})$. Thus, the distortion defined in (3) may be negative. We then reload the definition in this section as

$$D(t) = |N(t) - \hat{N}(t)|,\ t \geqslant 0, \tag{21}$$

where the operator $|\cdot|$ denotes the absolute value sign. The average distortion remains the definition in (4).

Since we are doing estimations for the previous sampling interval based on the current received sample, one may wonder the definition of real-time reconstruction. Hereby we clarify that the reconstruction is done in real-time as the way we introduce in Section II and Section III. However, we are still able to amend the curves that has been built in the past, just in case one may look back and be interested in the previous curve.

Except for *Algorithm* 1, there are many algorithms that can be carefully designed to further decrease the average distortion. However, these algorithms only decrease the sampling distortion since we are trying to recover the details omitted due to sampling. Thus, the total distortion cannot be eliminated.

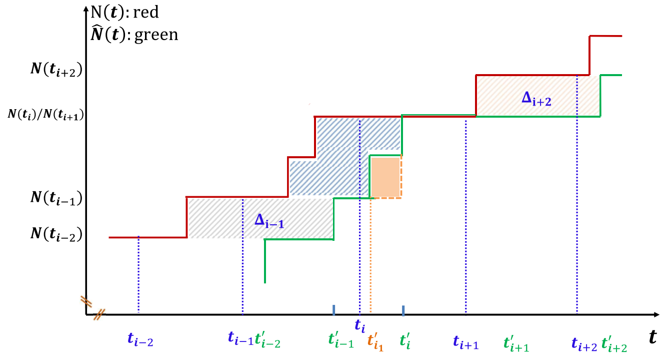


Fig. 8. Reconstruction with guessing: the orange area is the decreased distortion by adding one guessing point.

Algorithm 1 Reconstruct $\hat{N}(t)$ With Interpolation

```
Require:
    the time horizon $T$;
Ensure:
    the reconstructed process $\hat{N}(t)$;
Initialize: $\hat{N}(0) = 0$;
while receives a packet at time $t_i^{'}$ and $t_i^{'} < T$ do
    Sets $\hat{N}(t_i^{'}) = N(t_i)$ and calculates $J = \hat{N}(t_i^{'}) - \hat{N}(t_{i-1}^{'}) - 1$;
    if $J > 0$ then
        uniformly generates $J$ real numbers $\{t_{i_j}^{'} | 1 \leqslant j \leqslant J\}$ from open interval $(t_{i-1}^{'}, t_i^{'})$;
        for every $j$ in set $\{1, 2, \cdots, J\}$ do
            $\hat{N}(t_{i_1}^{'}) = \hat{N}(t_{i-1}^{'}) + j$;
        end for
    end if
    $i = i + 1$;
end while
```

B. The Lower Bound of the Average Distortion With Interpolation for Uniform Sampling Policy

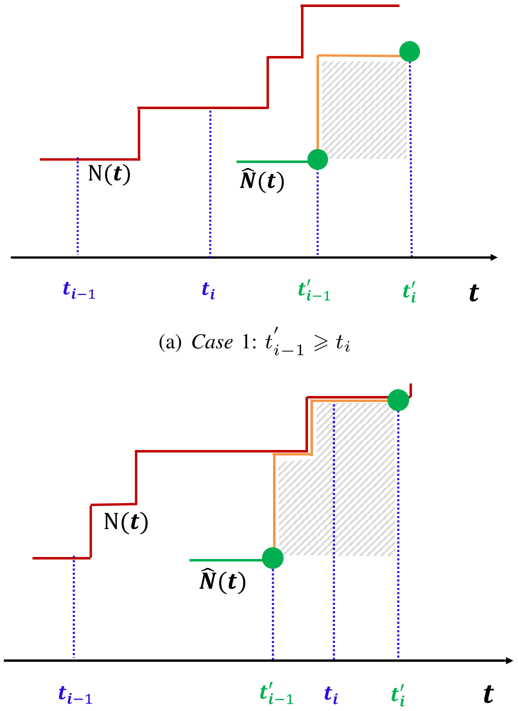
For We then give the upper bound of the performance of these interpolation algorithms for uniform sampling policy, namely, the lower bound of the average distortion.

Theorem 4: The lower bound of the average distortion under interpolation algorithms is given as

$$\breve{\Theta} = r\sigma\mu(1-\sigma)\mathrm{e}^{-\frac{\lambda}{r}} \sum_{n=1}^{\infty} \frac{\lambda^n}{n!r^n} \Bigg(n \int_{\frac{1}{r}}^{\infty} (x - \frac{1}{2r}) \mathrm{e}^{-\mu(1-\sigma)x} \mathrm{d}x + \frac{1}{2} \int_0^{\frac{1}{r}} x \mathrm{e}^{-(\lambda+\mu-\mu\sigma)x} \sum_{m=1}^{\infty} \frac{(\lambda x)^m}{(m-1)!} \mathrm{d}x \Bigg). \tag{22}$$

Proof: The principle that achieves $\breve{\Theta}$ is that, making the curve after interpolation approach the original process as much as possible such that the distortion area is decreased to the most extent. As shown in Fig. 9, this principle requires different operations for different cases.

Case 1: The received time of the $(i-1)$th sample $t_{i-1}^{'}$ and the sampling time of the ith sample t_i meet the requirement



(a) *Case* 1: $t'_{i-1} \geqslant t_i$

(b) *Case* 2: $t'_{i-1} < t_i$

Fig. 9. Sketches for *Case* 1 and *Case* 2.

$t'_{i-1} \geqslant t_i$. The minimal average area of one polygon in this case is calculated as[6]

$$\hat{S}_1 = \sum_{n=1}^{\infty} n\Pr\{N(d)=n\}\int_d^{\infty} p(x)(x-\frac{d}{2})\mathrm{d}x, \tag{23}$$

where $p(x)$ is the $p.d.f$ of the random variable defined as the sum of the waiting time and the service time, and given as [22]

$$p(x) = \sigma\mu(1-\sigma)\mathrm{e}^{-\mu(1-\sigma)x}, \; x \geqslant 0. \tag{24}$$

In *case* 1, the best reconstruction occurs if we increase the constructed process as soon as possible.

Case 2: The received time of the $(i-1)$th sample t'_{i-1} and the sampling time of the ith sample t_i meet the requirement $t'_{i-1} < t_i$. The minimal average area of one polygon in this case is calculated as

$$\hat{S}_2 = \sum_{n=1}^{\infty}\Pr\{N(d)=n\}\int_0^d \frac{x}{2}\sum_{m=1}^{n} m\Pr\{N(x)=m\}p(x)\mathrm{d}x. \tag{25}$$

In *case* 2, the best reconstruction occurs if we catch up with the original process as soon as possible and then keep pace with it.

Combining the two cases, we can obtain $\breve{\Theta} = r(\hat{S}_1 + \hat{S}_2)$ as

$$\breve{\Theta} = r\sum_{n=1}^{\infty}\Pr\{N(d)=n\}\left(n\int_d^{\infty} p(x)(x-\frac{d}{2})\mathrm{d}x \right. \\ \left. +\frac{1}{2}\int_0^d xp(x)\sum_{m=1}^{n} m\Pr\{N(x)=m\}\mathrm{d}x\right) \tag{26}$$

as the lower bound for average distortion when interpolation is adopted during reconstruction. Recall that, $\Pr\{N(d)=n\} = \frac{(\lambda d)^n}{n!}\mathrm{e}^{-\lambda d}$, $\Pr\{N(x)=m\} = \frac{(\lambda x)^m}{m!}\mathrm{e}^{-\lambda x}$, and the $p(x)$ in (24), we arrive at the conclusion in (22). ■

The specific strategy that achieves lower bound $\breve{\Theta}$ is ideal and impractical, however, it can be used to evaluate any given interpolation algorithm.

V. Simulation Verification

We present simulation results to validate the theoretical expressions of the derived average distortion under various sampling policies as well as the performance of the proposed algorithm. The simulation horizon T is set to 10^6.

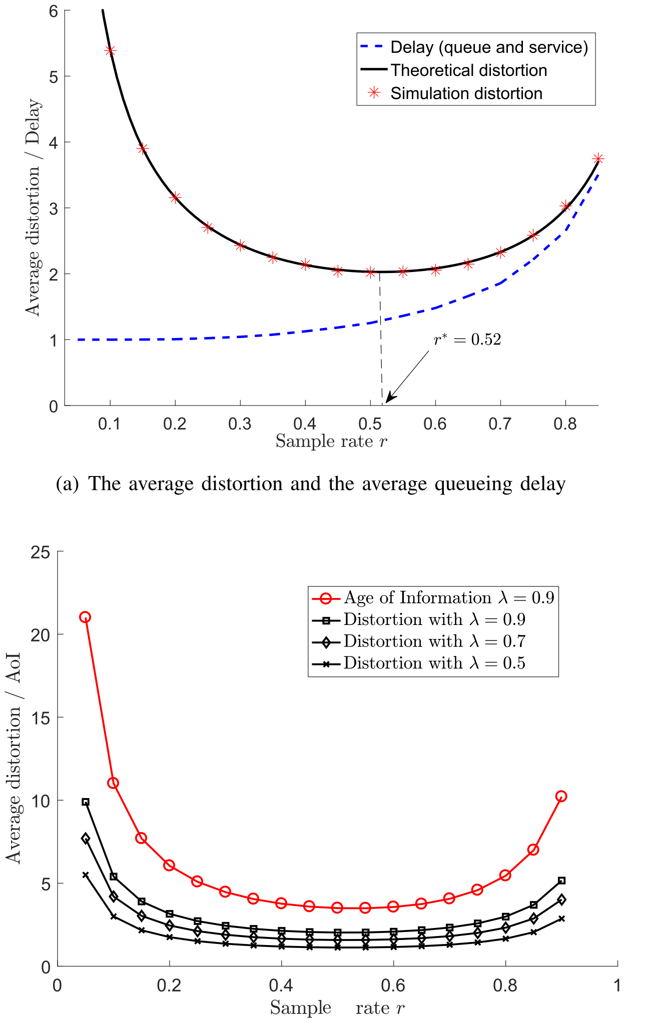
A. The Uniform Sampling Policy

In Fig. 10, we present the simulation and theoretical results for the uniform sampling method. In Fig. 10(a), we first plot the theoretical and the simulation results of the average distortion, as given in the black line and the line marked with red stars, respectively. We can see that, the simulation results are in accord with the theoretical results, which validates the accuracy of the average distortion-sampling rate function $\Theta(r)$ in (7). What's more, the tendency of the curve fits our expectation, i.e., it first decreases as the sampling rate increases due to more samples of the original signal are available by the receiver for the reconstruction, then increases due to that plentiful samples cause high waiting delay of the queueing system. Second, we also plot the average delay (the queueing delay plus the service time) experienced by the samples as given in the blue dash line. We can see that, the average delay increases as the sampling rate increases and the increasing is dramatical when the sampling rate is high. As a result, this dramatical increase causes the increase of average distortion. It shows that the defined average distortion reflects not only the traditional sampling distortion but also the transmission delay. It is a result of the tradeoff between the sampling and the transmission during the reconstruction.

Once we validate the correctness of the average distortion-sampling rate function, it can be used to find the optimal sampling rate which brings the minimum average distortion. For the adopted parameters in Fig. 10(a), the optimal sampling rate is obtained approximately as 0.52, which means that if we sample with this rate, the receiver can construct the best approximate of $N(t)$ in average.

In Fig. 10(b), we compare the average distortion and the AoI (age of information). As the sampling rate increases, they both first decrease and then increase. However, the AoI is only related to the sampling rate, which means that the AoI is a fix value for different processes. In our case, it means

[6] The derivations for conclusions in (23) and (25) are given in *Appendix* D due to space limitation.



(a) The average distortion and the average queueing delay

(b) The average distortion and the age of information

Fig. 10. Simulation results for the average sampling policy: the arrival rate of $N(t)$ for Fig. 10(a) is set to $\lambda = 0.9$ and the average service rate for both Fig. 10(a) and Fig. 10(b) is set to $\mu = 1$.

for counting process with different arrival rates λ, they have same AoI. The average distortion for different arrival rates induce different curves. Actually, the average distortion and the AoI under uniform sampling policy have the relationship that $\Theta(r) = \lambda\Delta$, where Δ represents the AoI.[7] Thus, by setting the arrival rate to $\lambda = 1$, the average distortion curves and the AoI curves will overlap as shown in Fig. 11. It means that for Poisson process, the defined average distortion collapses to the AoI metric.

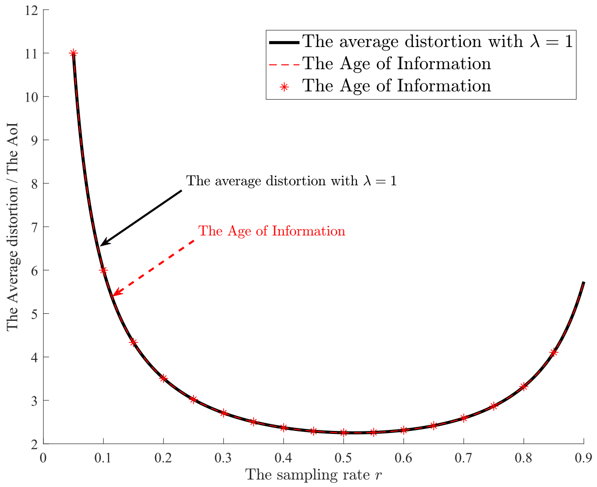


Fig. 11. The average distortion curve and the AoI curve overlap with each other when $\lambda = 1$.

B. The Threshold-Based Policy

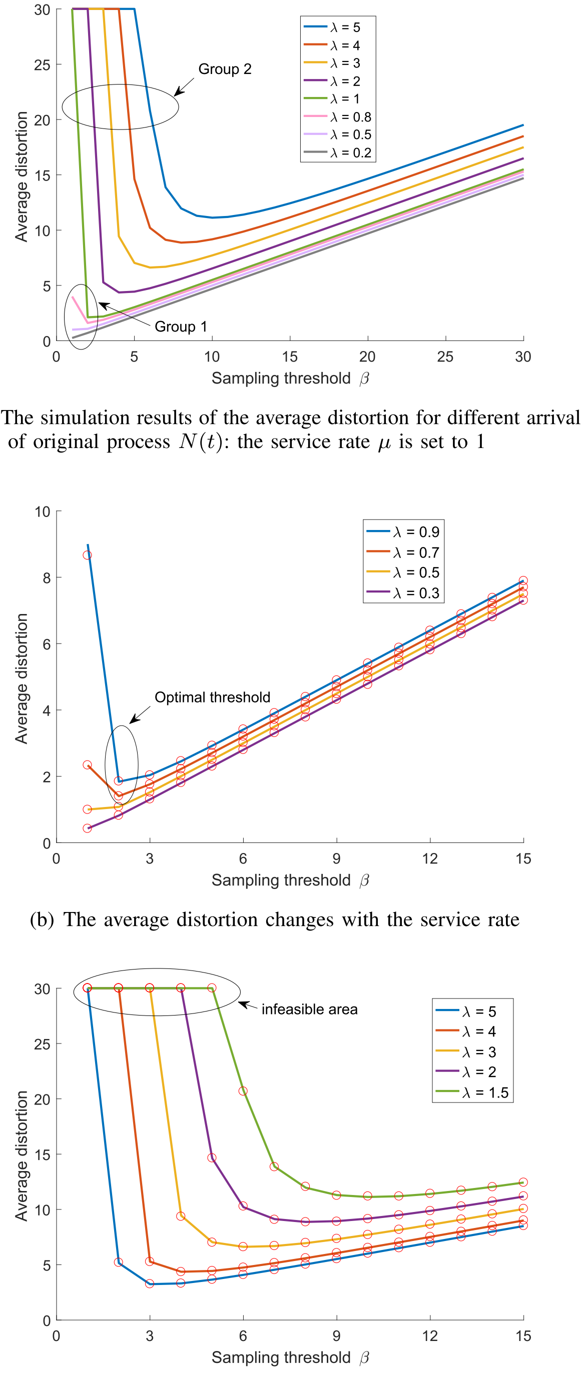
We give the simulation results for the threshold-based sampling method in Fig. 12. Firstly, with the service rate μ being 1, we give the simulation results of the average distortion for a range of arrival rate λ of the original process in Fig. 12(a). We can see that, these curves can be divided into two groups as the circles. The first group is when the arrival rate is below than $\mu = 1$. In this case, the queueing system is stable and the average distortion is finite even when the threshold is 1, i.e., any update of $N(t)$ will be recorded and transmitted which brings heavy traffic to the queue. The other group is when the arrival rate $\lambda \geqslant 1$. In this case, the heavy traffic bought by the small threshold makes the queue system unstable and leads to infinite average distortion. We set a great number (it is 30 in the figure) if the queue is unstable. With the threshold increasing, the queueing system becomes stable.

The theoretical and simulation results for the two groups are given in Fig. 12(b) and Fig. 12(c), respectively. The accordance validates the expression of the average distortion given in (15). In Fig. 12(b), increasing the threshold β when sampling may not bring any benefits for smaller λ, as shown in the cases $\lambda = 0.3$ and $\lambda = 0.5$. For greater arrival rates, such as the cases $\lambda = 0.7$ and $\lambda = 0.9$, the systems works in a relative high load mode for threshold $\beta = 1$. Thus, increasing the threshold a little bit greater to $\beta = 2$ will decrease the average distortion distinctly. Thus, the optimal threshold $\beta^* = 1$ for smaller λ and $\beta^* = 2$ for greater λ. In Fig. 12(c), there exist infeasible thresholds for all cases as shown in the left upper area of the figure. For this group, the optimal thresholds vary for different arrival rates. Using the theoretical average distortion, it is easily to derive the optimal threshold value as 3, 4, 6, 8, and 10 for $\lambda = 1.5$, 2, 3, 4, and 5, respectively.

C. The Zero-Wait Sampling Policy

We give the simulation and theoretical results of the average distortion under the zero-wait sampling policy in Fig. 13. Considering that the zero-wait policy is a fix policy, the average distortion is a constant value when the arrival rate of $N(t)$ and the service rate are given. We plot the distortion-arrival rate and the distortion-service rate curves in Fig. 13(a) and Fig. 13(b), respectively. In both cases, the correctness of the average distortion given in (20) can be validated, since the simulation and the theoretical results are in agreement. Specifically, in Fig. 13(a), for a fixed service rate, the average distortion increases linearly with the increasing of arrival rate of $N(t)$. In Fig. 13(b), for a fixed arrival rate, the average

[7] In [18], the AoI is given as $\Delta = \frac{1}{\mu}(\frac{1}{2\rho} + \frac{1}{1-\beta})$, where $\rho = \frac{1}{\mu d}$. With a little transformation, we can find $\Theta(r) = \lambda\Delta$.



(a) The simulation results of the average distortion for different arrival rate of original process $N(t)$: the service rate μ is set to 1

(b) The average distortion changes with the service rate

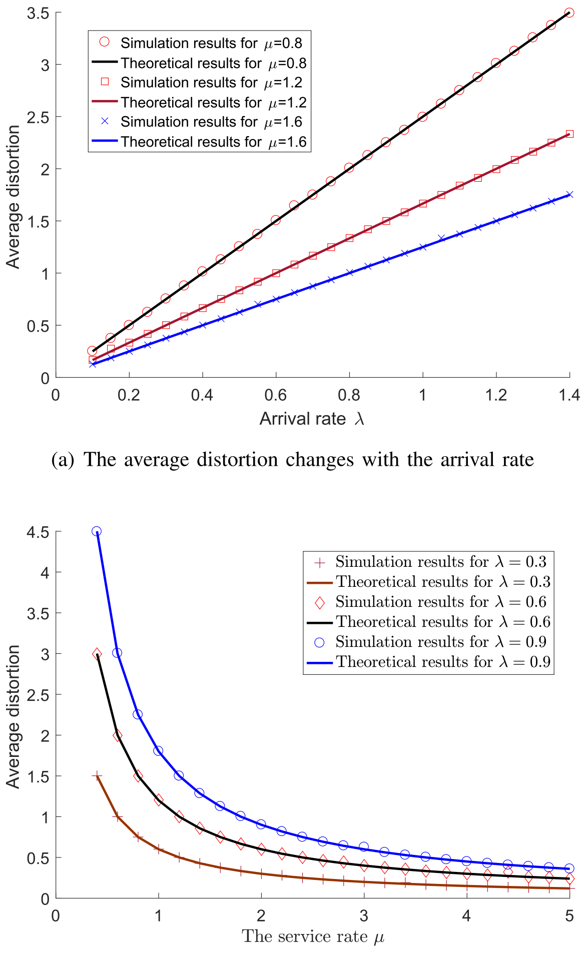
(c) The average distortion changes with the service rate

Fig. 12. Simulation results for the threshold-based sampling policy.

distortion decreases inverse proportionally with the increasing of the service rate μ.

D. The Comparison of the Three Sampling Policies

In Fig. 14, we give the comparison results of the three sampling polices. The service rate of the queue is fixed as $\mu = 1$. We plot the curves that the minimum average distortion versus the arrival rate. The minimum average distortion is obtained by reconstructing the process with the optimal sampling rate or threshold for the uniform policy or the threshold-based policy. We can see that, the zero-wait policy performs almost the best among the three policies. The reason is that the sampled measurement can be served and received directly after being generated, instead of waiting in the queue and making the measurement become less fresh. This result is in accord with [23], where the authors validated the conclusion that it can



(a) The average distortion changes with the arrival rate

(b) The average distortion changes with the service rate

Fig. 13. Simulation results for the zero-wait sampling policy.

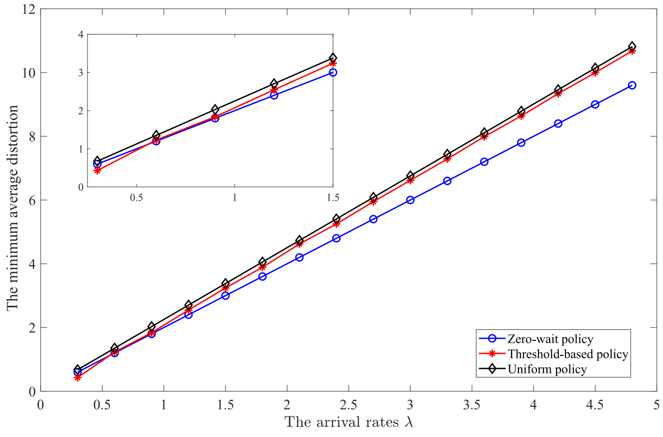


Fig. 14. The comparison of the uniform policy, the threshold-based policy, and the zero-wait policy: the service rate of the queue is set to $\mu = 1$.

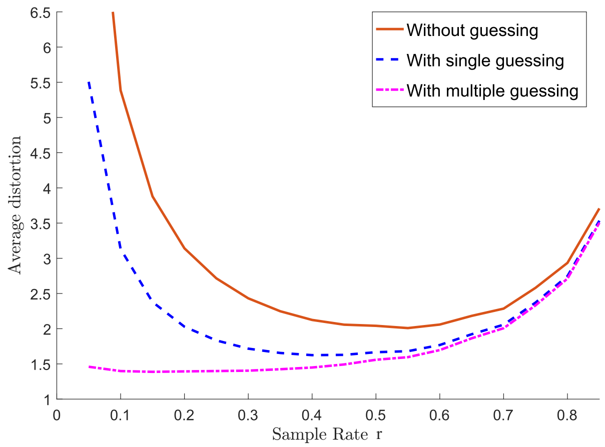


Fig. 15. The simulation results for the proposed algorithm: the monitor is able to make some guesses for the omitted details based the received samples.

be better for updating source to be lazy rather than deliver relatively non-information updates. The threshold-based policy outperforms the uniform sampling policy but the difference is slight. The total sampling time for the uniform policy is r^*T and for the threshold-based policy is $\frac{\lambda}{\beta^*}T$. From the numerical results we know that $\frac{\lambda}{\beta^*} < r^*$, it means the uniform policy transmits more sample packets to the monitor compared to the threshold-based policy. Thus, the greater traffic load of the queue induces the greater distortion for uniform policy.

E. Reconstruction With Interpolation

In Fig. 15, we give the simulation results of the proposed algorithm. The simulation parameters are set as same as the parameters in Fig. 10. The orange solid line is the average distortion when the reconstruction is based only on the received samples. We can see that, for smaller sampling rate, the average distortion is very high since many details of the counting process are omitted and just finite snapshots of the original process are available to the monitor. The blue dash line is the average distortion obtained when only one guessing point is added to the reconstructed process. We can see that, the average distortion is decreased compared to the no guessing case, especially for small sampling rate when the sampling distortion contributes more to the average distortion. The purple dash line is the average distortion obtained exactly based on *Algorithm* 1, where multiple guessing points are added and these points are distributed uniformly between two received samples. The average distortion can be further decreased compared to the blue line. Especially, when the sample rate r approaches zero, the average distortion is finite instead of approaching infinity as the other two cases.[8] With the idea of adding guessing points, the sampling distortion can be decreased but it improves nothing for the transmission distortion. Thus, neither the blue nor purple lines improve the performance for higher sampling rate.

VI. Conclusion

We consider the real-time remote reconstruction of a counting process, where distortion occurs due to the sampling and the queueing behaviors. The average distortion defined as the average gap between the original signal and the reconstructed signal is adopted to measure the performance of the reconstruction. We study the uniform sampling method and two non-uniform sampling policies, i.e., the threshold-based policy and the zero-wait policy. For all the three policies, we derive the theoretical expressions of the average distortion as functions of some key sampling parameters. The theoretical results help us find the optimal sampling parameters that balance the tradeoff between the sampling distortion and the distortion caused by queueing, and then induce the minimum average distortion. We also introduce interpolation algorithms that can further decrease the average distortion, and derive the upper bound of the performance of these algorithm for uniform sampling policy. Simulation results validate our derivation and at the mean time provide a way to understand the defined average distortion. Future works include pursuing theoretical results for other counting processes with other sampling method and packet scheduling policies.

Appendix A
The Finite Packet Length of the Sample

For uniform sampling, samples are obtained at time instants $\{t_i = id, i \geqslant 1\}$. It means that one sample should be able convey the increase of the original process during one sample interval d, i.e., $N(d)$. For the Poisson process, we know that

$$\Pr\{N(d) = n\} = \frac{(\lambda d)^n}{n!}e^{-\lambda d}, \quad n \in \mathbb{N}. \tag{27}$$

Suppose that the length of one packet that used to convey the raw information is k bits. Then, the effective range of one packet is $\{0, 1, \cdots, M = 2^k - 1\}$. Once $N(d)$ exceeds this range, the sample fails to exactly convey the increase. The probability of failure can be calculated as

$$\varrho = \sum\nolimits_{i=M}^{\infty} \Pr\{N(d) = n\}. \tag{28}$$

Failure probability ϱ decreases with the increase of k. However, a greater k will lead to waste, especially when $N(d)$ is small. In this work, we ideally assume that any $N(d)$ can be exactly expressed by one sample.

Appendix B
The Calculation of $\mathbb{E}\{S_A\}$, $\mathbb{E}\{S_B\}$, and $\mathbb{E}\{S_C\}$ in (9)

In this section, we given $\mathbb{E}\{S_A\}$ in *Lemma* 7, $\mathbb{E}\{S_B\}$ and $\mathbb{E}\{S_C\}$ in *Lemma* 8.

Lemma 7: Area S_A is the distortion caused by sampling and its average is given as

$$\mathbb{E}\{S_A\} = \frac{\lambda}{2r^2}. \tag{29}$$

[8] In detail, as the sampling rate approaches zero, it means the monitor merely knows the measurements of $N(t)$ at finite sporadic sampling instants. According to *Theorem* 2.3.1 in reference [24], given interval (t_{start}, t_{end}) and $n = N(t_{end}) - N(t_{start})$, the arrival times of the n events have the same distributions as the order statistics corresponding to n *i.i.d.* random variables uniformly distributed on the interval. Thus, the distortion caused by sampling can be eliminated statistically. This conclusion holds only for Poission process and uniformly inserting estimations.

Proof: Let $N(d)$ denote the number of the events that happen during sampling interval d, i.e., the height of the polygon, we can obtain S_A as

$$S_A = \sum_{k=1}^{N(d)} (d - s_k). \tag{30}$$

Clearly, the average area of A, i.e., $\mathbb{E}\{S_A\}$, is dependent on the stochastic property of the original counting process $N(t)$. For Poisson counting process, the heights of Δ_i, $i \geqslant 1$ are $i.i.d$ stochastic Poisson variables with parameter λd. In order to calculate $\mathbb{E}\{S_A\}$, we first calculate the conditional average when $N(d)$ is given as[9]

$$\begin{aligned} \mathbb{E}\{S_A|N(d)=n\} &\overset{\textcircled{1}}{=} nd - \mathbb{E}\left\{\sum\nolimits_{k=1}^{n} s_k|N(d)=n\right\} \quad (31)\\ &\overset{\textcircled{2}}{=} nd - \mathbb{E}\left\{\sum\nolimits_{k=1}^{n} Y_k\right\}\\ &= nd - \frac{nd}{2} = \frac{nd}{2}, \end{aligned}$$

where step ① holds because of the condition $N(d)=n$ and step ② holds due to *Theorem* 5 in *Appendix* C. Thus, we can obtain $\mathbb{E}\{S_A\}$ as

$$\begin{aligned} \mathbb{E}\{S_A\} &= \mathbb{E}\left\{\mathbb{E}\left\{S_A|N(d)=n\right\}\right\}\\ &= \sum\nolimits_{n=0}^{\infty} \Pr\{N(d)=n\}\mathbb{E}\{S_A|N(d)=n\} \quad (32)\\ &= \sum\nolimits_{n=0}^{\infty} \Pr\{N(d)=n\}\frac{nd}{2}\\ &= \frac{d}{2}\mathbb{E}\{N(d)\} = \frac{\lambda d^2}{2}. \end{aligned}$$

Considering the relationship between d and r, i.e., $r = \frac{1}{d}$, we arrive at the conclusion in (29). ■

Lemma 8: The sum of the average areas of sub-polygons B and C is given as

$$\mathbb{E}\{S_B + S_C\} = \frac{\lambda}{r}\left(\frac{\sigma}{\mu(1-\sigma)} + \frac{1}{\mu}\right). \tag{33}$$

Proof: Sub-polygons B and C are rectangles, whose areas can be calculated as $S_B = N(d)w_i$ and $S_C = N(d)v_i$, respectively, where w_i is the queueing delay and v_i is the service time of the ith sample packet. With uniform sampling, the queueing model is formulated as a $D/M/1$ model. Thus the average queueing delay and the average service time have already been calculated [24] and given as $\mathbb{E}\{w_i\} = \frac{\sigma}{\mu(1-\sigma)}$ and $\mathbb{E}\{v_i\} = \frac{1}{\mu}$, respectively, where the parameter σ is the solution of Lambert W function given in (8). Given sampling rate r, $N(d)$ is independent with w_i and v_i. Thus, we obtain

$$\begin{aligned} \mathbb{E}\left\{S_B + S_C\right\} &= \mathbb{E}\left\{N(d)\right\}\Big(\mathbb{E}\left\{w_i\right\} + \mathbb{E}\left\{v_i\right\}\Big)\\ &= \lambda d\left(\frac{\sigma}{\mu(1-\sigma)} + \frac{1}{\mu}\right). \quad (34) \end{aligned}$$

With $r = \frac{1}{d}$, we arrive at the conclusion in (33). ■

APPENDIX C
THE PROOF OF STEP ② IN (31)

In *Theorem* 5, we give the detailed proof of Step ② in (31).

Theorem 5: Given $N(d) = n$, $\{Y_k, 1 \leqslant k \leqslant n\}$ are independent random variables that are uniformly distributed on interval $(0,d)$, we have

$$\mathbb{E}\left\{\sum\nolimits_{k=1}^{n} s_k|N(d)=n\right\} = \mathbb{E}\left\{\sum\nolimits_{k=1}^{n} Y_{(k)}\right\} = \frac{nd}{2}. \tag{35}$$

Proof: Considering s_k is the occur times of the kth event and given $N(d) = n$ means there already have n events occurred, then the occur time of these n packets must meet the requirement that $s_1 \leqslant s_2 \leqslant \cdots \leqslant s_k \leqslant \cdots \leqslant s_n$. Based on the conclusion for Poisson process that given $N(d) = n$, the occur times s_k, $1 \leqslant k \leqslant n$ have the same distribution of $Y_{(k)}$s, where $Y_{(k)}$ is the kth smallest value among Y_k [25]. Thus, we have

$$\mathbb{E}\left\{\sum\nolimits_{k=1}^{n} s_k|N(d)=n\right\} = \mathbb{E}\left\{\sum\nolimits_{k=1}^{n} Y_{(k)}\right\}. \tag{36}$$

In the right hand of (36), the sequence of variables do not matter since we are calculating the sum. Thus, we have

$$\mathbb{E}\left\{\sum\nolimits_{k=1}^{n} Y_{(k)}\right\} = \mathbb{E}\left\{\sum\nolimits_{k=1}^{n} Y_k\right\}. \tag{37}$$

Remember that, Y_k is a random variable that uniformly distributed on interval $(0,d)$, we have $\mathbb{E}[Y_k] = \frac{d}{2}$. Thus,

$$\mathbb{E}\left\{\sum\nolimits_{k=1}^{n} Y_k\right\} = \sum\nolimits_{k=1}^{n} \mathbb{E}\{Y_k\} = \frac{nd}{2}. \tag{38}$$

By far, we have completed the proof. ■

APPENDIX D
DETAILED CALCULATION FOR (23) AND (25)

In this section, we derive the expressions given in (23) and (25).

We first calculate (23) for case 1. Recall that, $p(x)$ is the $p.d.f.$ of variable $x_i = w_i + v_i$. For case 1, $t_{i-1}^{'} \geqslant t_i$. Thus, given the condition that there are n events occurring during the sampling interval, the area of the polygon can be drawn as

$$\varphi = \int_{d}^{\infty} p(x) \sum_{m=1}^{n} (x - s_n)\mathrm{d}x. \tag{39}$$

Combining the result in *Theorem* 5, we know

$$\begin{aligned} \mathbb{E}[\varphi] &= \int_{d}^{\infty} p(x)\mathbb{E}\left\{\sum_{m=1}^{n}(x - s_n)\right\}\mathrm{d}x\\ &= \int_{d}^{\infty} p(x)\left(nx - \frac{nd}{2}\right)\mathrm{d}x. \quad (40) \end{aligned}$$

Thus, $\hat{S}_1$ can be calculated as

$$\begin{aligned} \hat{S}_1 &= \sum_{n=1}^{\infty} \Pr\{N(d)=n\}\mathbb{E}\{\varphi\}\\ &= \sum_{n=1}^{\infty} n\Pr\{N(d)=n\}\int_{d}^{\infty} p(x)(x - \frac{d}{2})\mathrm{d}x. \quad (41) \end{aligned}$$

[9] We define $\{Y_k, 1 \leqslant k \leqslant n\}$ are independent random variables that are uniformly distributed on interval $(0,d)$, see *Theorem* 5 in *Appendix* C.

We then calculate (25) for case 2. For case 2, $t_{i-1}^{'} < t_i$. Thus, given the condition that there are n events occurring during the sampling interval, the area of the polygon can be drawn as

$$\varsigma = \int_0^d p(x) \sum_{m=1}^{n} \Pr\{N(x) = m\} \sum_{k=1}^{m} (x - s_k)\mathrm{d}x. \tag{42}$$

With the conclusion in *Theorem* 5, the expectation of ς is calculated as

$$\begin{aligned} \mathbb{E}[\varsigma] &= \int_0^d p(x) \sum_{m=1}^{n} \Pr\{N(x) = m\} \mathbb{E}\left\{ \sum_{k=1}^{m} (x - s_k) \right\} \mathrm{dx} \\ &= \int_0^d p(x) \sum_{m=1}^{n} \Pr\{N(x) = m\} \frac{mx}{2} \mathrm{dx}. \end{aligned} \tag{43}$$

Thus, $\hat{S}_2$ can be calculated as

$$\begin{aligned} \hat{S}_2 &= \sum_{n=1}^{\infty} \Pr\{N(d) = n\} \mathbb{E}\{\varsigma\} \\ &= \sum_{n=1}^{\infty} \Pr\{N(d) = n\} \int_0^d \frac{x}{2} \sum_{m=1}^{n} m\Pr\{N(x) = m\} p(x) \mathrm{dx}. \end{aligned} \tag{44}$$

Appendix E
The Calculation of $\mathbb{E}\{S_A\}$ and $\mathbb{E}\{S_B\}$ in *Lemma* 4

Lemma 9: The area of sub-polygon A is denoted by S_A, whose average is obtained as

$$\mathbb{E}\{S_A\} = \frac{(\beta - 1)\beta}{2\lambda}. \tag{45}$$

Proof: We first express the area of sub-polygon A as

$$S_A = \sum_{n=1}^{\beta} (s_\beta - s_n), \tag{46}$$

where s_β is the total time that needed for β events occurring and $\{s_n,\ n = 1, 2, \cdots, \beta\}$ are the occur time of the nth event. For Poisson counting process, we know that variable s_β submits to an Erlang distribution with stage β and mean $\frac{r}{\lambda}$. Its $p.d.f.$ is given as

$$h(x) = \frac{\lambda^\beta x^{\beta-1} e^{-\beta x}}{(\beta - 1)!},\ x \geqslant 0. \tag{47}$$

We then calculate the conditional expectation $\mathbb{E}\{S_A | s_\beta = x\}$, i.e.,

$$\mathbb{E}\{S_A | s_\beta = x\} = \sum_{n=1}^{\beta} (x - s_n) = \sum_{n=1}^{\beta-1} (x - s_n), \tag{48}$$

where the last equation holds due to the fact that $x = s_n$ for the term $n = \beta$. Equation (48) is calculated under the condition that there are $(\beta - 1)$ events occur in sequence during interval $[0, x)$. Recall the conclusion in *Theorem* 5 and let $d = x$ and $n = \beta - 1$, we have the following conclusion

$$\mathbb{E}\{S_A | s_\beta = x\} = \frac{(\beta - 1)x}{2}. \tag{49}$$

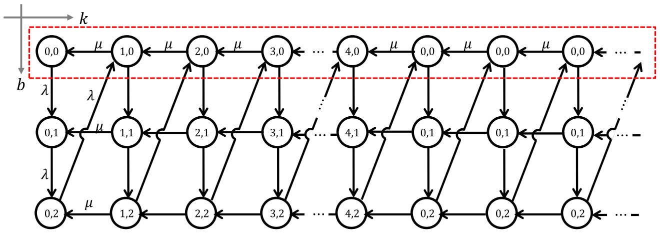


Fig. 16. The transitions between joint queue-phase states when the threshold β is given as 3. The horizontal axis is the queue length, i.e., the number of the packets queueing in the buffer while the vertical axis is the arrival phase, i.e., the number of the events that occur since last sampling. The states in the red dash box are the states that observed by new generated sample.

Then, we are able to calculate $\mathbb{E}\{S_A\}$ as

$$\begin{aligned} \mathbb{E}\{S_A\} &= \int_0^\infty \mathbb{E}\{S_A | s_\beta = x\}\ h(x)\mathrm{dx} \\ &= \int_0^\infty \frac{(\beta - 1)x}{2}\ h(x)\mathrm{dx} \\ &= \frac{(\beta - 1)\beta}{2\lambda}. \end{aligned} \tag{50}$$

Thus, we arrive at the conclusion in (45). ■

Lemma 10: The area of sub-polygon B is denoted by S_B, whose average is given as

$$\mathbb{E}\{S_B\} = \beta \mathbb{E}\{w_i\} = \frac{\beta}{\mu(z_0^\beta - 1)}. \tag{51}$$

Proof: Sub-polygon B is a rectangle, whose area is calculated as

$$\mathbb{E}\{S_B\} = \beta \mathbb{E}\{w_i\}, \tag{52}$$

where w_i is the queueing delay of the packets. To obtain $\mathbb{E}\{w_i\}$, we turn back to the system model in Fig. 5(a), where the queueing system can be modeled as an $E_r/M/1$ queueing system. For such a queueing system, it is not adequate to just calculate the steady-state probability of the queue states π_k, namely, the probability that there are k sample packets in the queue. Because the packet arrival process has β phases even for the same queue state $'queue\ length = k'$, corresponding to the state that there already have b $(0 \leqslant b \leqslant \beta - 1)$ events occurred when there are k samples queueing in the buffer. By defining the joint steady-state probability as $\pi_{k,b} = \Pr\{queue\ length = k,\ phase = b\}$ and abbreviating it to $\pi_{k,b} = \Pr\{ql = k,\ ph = b\}$, we have the following transition probabilities

$$\begin{cases} \Pr\{ql = k,\ ph = b + 1 \mid ql = k,\ ph = b\} = \lambda, \\ \hfill k \geqslant 0,\ 0 \leqslant b < \beta - 1; \\ \Pr\{ql = k + 1,\ ph = 0 \mid ql = k,\ ph = \beta - 1\} = \lambda, \\ \hfill k \geqslant 0; \\ \Pr\{ql = k - 1,\ ph = b \mid ql = k,\ ph = b\} = \mu, \\ \hfill k \geqslant 1,\ 0 \leqslant b \leqslant \beta - 1. \end{cases}$$

In Fig. 16, an example of the transitions between the joint queue-phase states is given with the threshold for the sampling being 3. As time goes by, one can trace the system state in a zigzag way. For example, starting from state $(0,0)$ which means there is no sample in the queue and no event occurs, the system jumps to state $(0,1)$ with a new event occurring. Since the accumulated change of the original process is less than the threshold $\beta = 3$, no sample is generated and the queue state remains as zero. The system then jumps to state $(0,2)$ with another event occurring and still no sample being generated. At state $(0,2)$, one event occurring will increase the phase state to 3 which equals to threshold $\beta = 3$. Thus, a new sample is generated and enters the queue. This makes the queue state increase to 1 and the phase state reset to zero, and so on.

However, we do not care all of joint queue-phase states of the system shown in Fig. 16. Only the states observed by the new generated samples when they enter the queue influence the waiting time, as shown in the red dash box in Fig. 16. Using the global equilibrium equations and the normalization equation, we can calculate the joint queue-phase steady-state probability $\{\pi_{k,b} \mid k \geqslant 0, 0 \leqslant b < \beta\}$, among which, the states observed by the new samples are given as

$$\pi_{k,0} = \frac{\lambda}{\beta\mu}(z_0 - 1){z_0}^{-k\beta-1}, \ k \geqslant 0. \tag{53}$$

Since $\{\pi_{k,0}, \ k \geqslant 0\}$ are only part of the whole states, the sum of $\pi_{k,0}$ is less than one. Thus, the steady-state probabilities should be well normalized. Let Π be the sum of $\{\pi_{k,0}, \ k \geqslant 0\}$, namely,

$$\Pi = \sum_{k=0}^{\infty} \pi_{k,0} = \frac{\lambda(z_0 - 1)}{\beta\mu z_0(1 - {z_0}^{\beta})}. \tag{54}$$

We then normalize the probabilities as

$$\pi_{k,0}^{*} = \frac{\pi_{k,0}}{\Pi} = {z_0}^{-k\beta}(1 - {z_0}^{-\beta}). \tag{55}$$

With $\pi_{k,0}^{*}$, we can calculate the average queueing length of the queue $\mathbb{E}\{q\}$ and the average waiting time $\mathbb{E}\{w\}$ as

$$\mathbb{E}\{q\} = \sum_{k=0}^{\infty} k\pi_{k,0}^{*} = \frac{1}{z_0^{\beta} - 1}, \tag{56}$$

$$\mathbb{E}\{w\} = \frac{\mathbb{E}\{q\}}{\mu} = \frac{1}{\mu(z_0^{\beta} - 1)}, \tag{57}$$

respectively. Replacing $\mathbb{E}\{w\}$ in (52) with the result in (57), we arrive at (51). ■

Appendix F
The Calculation of $\mathbb{E}\{S_A\}$ and $\mathbb{E}\{S_B\}$ in *Lemma* 6

Lemma 11: The area of A is denoted by S_A and the average of S_A is given as

$$\mathbb{E}\{S_A\} = \frac{\lambda}{\mu^2}. \tag{58}$$

Proof: The area of sub-polygon A can be obtained as

$$S_A = \sum_{k=1}^{N(v_{i-1})} (v_{i-1} - s_k), \tag{59}$$

where v_{i-1} is the service time of sample obtained at t_{i-1} and $N(v_{i-1})$ is the number of events occurring during time span $v_{i-1} = t_i - t_{i-1}$. In order to calculate the expectation of S_A, we first calculate

$$\mathbb{E}\{S_A | v_{i-1} = d\} = \mathbb{E}\left\{\sum_{k=1}^{N(d)} (d - s_k)\right\}. \tag{60}$$

Note that, the part in the bracket of the right hand of (60) is exactly the same as (30). Thus, we can use the conclusion obtained in (32), i.e,

$$\mathbb{E}\{S_A | v_{i-1} = d\} = \frac{\lambda d^2}{2}. \tag{61}$$

Remember that, v_{i-1} is an exponential random variable with parameter μ. Thus, we calculate the expectation of S_A as follows.

$$\mathbb{E}\{S_A\} = \mathbb{E}\{\mathbb{E}\{S_A | v_{i-1} d\}\} \tag{62}$$

$$= \frac{\lambda}{2}\mathbb{E}\{d^2\} = \frac{\lambda}{2}\frac{2}{\mu^2} = \frac{\lambda}{\mu^2}. \tag{63}$$

Thus, we have arrived at the conclusion in (58). ■

Lemma 12: The area of sub-polygon B can be obtained as

$$\mathbb{E}\{S_B\} = \frac{\lambda}{\mu^2}. \tag{64}$$

Proof: Sub-polygon B is a rectangle, whose area is obtained as

$$S_B = N(v_{i-1})v_i. \tag{65}$$

Given the arrival rate of the original process, $N(v_{i-1})$ is independent with v_i. Thus, we have,

$$\begin{aligned} \mathbb{E}\{S_B\} &= \mathbb{E}\{N(v_{i-1})v_i\} \\ &= \mathbb{E}\{N(v_{i-1})\}\,\mathbb{E}\{v_i\} \\ &= \frac{\lambda}{\mu}\frac{1}{\mu} = \frac{\lambda}{\mu^2}. \end{aligned} \tag{66}$$

Thus, we have arrived at the conclusion in (64). ■

Acknowledgment

The authors would like to thank the Associate Editor and anonymous reviewers for their detailed comments and constructive suggestions that have much improved the presentation and quality of this article.

References

[1] M. Díaz, C. Martín, and B. Rubio, “State-of-the-art, challenges, and open issues in the integration of Internet of Things and cloud computing,” *J. Netw. Comput. Appl.*, vol. 67, pp. 99–117, May 2016.

[2] G. M. Dias, M. Nurchis, and B. Bellalta, “Adapting sampling interval of sensor networks using on-line reinforcement learning,” in *Proc. IEEE 3rd World Forum Internet Things (WF-IoT)*, Dec. 2016, pp. 460–465.

[3] S. Li, L. D. Xu, and X. Wang, “Compressed sensing signal and data acquisition in wireless sensor networks and Internet of Things,” *IEEE Trans Ind. Informat.*, vol. 9, no. 4, pp. 2177–2186, Nov. 2013.

[4] E. J. Candes and M. B. Wakin, “An introduction to compressive sampling,” *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[5] H. Nyquist, “Certain topics in telegraph transmission theory,” *Proc. IEEE*, vol. 90, no. 2, pp. 280–305, Feb. 2002.

[6] H. J. Landau,“Sampling, data transmission, and the Nyquist rate,” *Proc. IEEE*, vol. 55, no. 10, pp. 1701–1706, Oct. 1967.

[7] E. J. Candes, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[8] Y. C. Eldar, *Sampling Theory: Beyond Bandlimited Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2015.

[9] F. Marvasti, *Nonuniform Sampling: Theory and Practice*. Boston, MA, USA: Springer, 2012.

[10] N. Vaswani and J. Zhan, "Recursive recovery of sparse signal sequences from compressive measurements: A review," *IEEE Trans. Signal Process.*, vol. 64, no. 13, pp. 3523–3549, Jul. 2016.

[11] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[12] A. Maleki and D. L. Donoho, "Optimally tuned iterative reconstruction algorithms for compressed sensing," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 330–341, Apr. 2010.

[13] M. S. Asif and J. Romberg, "Dynamic updating for $\ell 1$ minimization," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 421–434, Apr. 2010.

[14] M. S. Asif and J. Romberg, "Sparse recovery of streaming signals using ℓ_1-homotopy," *IEEE Trans. Signal Process.*, vol. 62, no. 16, pp. 4209–4223, Aug. 2014.

[15] U. L. Wijewardhana and M. Codreanu, "A Bayesian approach for online recovery of streaming signals from compressive measurements," *IEEE Trans. Signal Process.*, vol. 65, no. 1, pp. 184–199, Jan. 2017.

[16] Y. Sun, Y. Polyanskiy, and E. Uysal-Biyikoglu, "Remote estimation of the Wiener process over a channel with random delay," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 321–325.

[17] A. M. Bedewy, Y. Sun, and N. B. Shroff, "Optimizing data freshness, throughput, and delay in multi-server information-update systems," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2016, pp. 2569–2573.

[18] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 2731–2735.

[19] C. Kam, S. Kompella, and A. Ephremides, "Age of information under random updates," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2013, pp. 66–70.

[20] M. Costa, M. Codreanu, and A. Ephremides, "On the age of information in status update systems with packet management," *IEEE Trans. Inf. Theory*, vol. 62, no. 4, pp. 1897–1910, Apr. 2016.

[21] C. Kam, S. Kompella, G. D. Nguyen, J. E. Wieselthier, and A. Ephremides, "Controlling the age of information: Buffer size, deadline, and packet replacement," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Nov. 2016, pp. 301–306.

[22] L. Kleinrock, *Queueing Systems. Theory*, vol. 1. New York, NY, USA: Wiley, 1975.

[23] R. D. Yates, "Lazy is timely: Status updates by an energy harvesting source," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2015, pp. 3008–3012.

[24] D. Gross, *Fundamentals of Queueing Theory*. New York, NY, USA: Wiley, 2008.

[25] S. M. Ross, *Stochastic Processes*. New York, NY, USA: Wiley, 1996.

Meng Wang (Student Member, IEEE) received the B.S. degree from Xidian University, Shaanxi, China, in 2014, and the Ph.D. degree from Tsinghua University, Beijing, China, in 2019. In 2017 and 2018, she was a Visiting Ph.D. Student with the University of Maryland. Her research interests include crosslayer design, delay-power tradeoff, constraint MDP, and the age of information (AoI).

Wei Chen (Senior Member, IEEE) received the B.S. and Ph.D. degrees (Hons.) from Tsinghua University in 2002 and 2007, respectively. From 2014 to 2016, he was the Deputy Head of the Department of Electronic Engineering, Tsinghua University. From 2005 to 2007, he was a Visiting Ph.D. Student with The Hong Kong University of Science and Technology. He visited the University of Southampton in 2010, Telecom Paris Tech in 2014, and Princeton University, Princeton, NJ, USA, in 2016. Since 2007, he has been a Faculty Member with Tsinghua University, where he is currently a Tenured Full Professor, the Director of the Degree Office, and a University Council Member. He has also been supported by the National 973 Youth Project, the NSFC Excellent Young Investigator Project, the New Century Talent Program of the Ministry of Education, and the Beijing Nova Program. His research interests are in the areas of communication theory, stochastic optimization, and statistical learning. He is a Cheung Kong Young Scholar and a member of the National Program for Special Support of Eminent Professionals, also known as 10 000 talent program. He is also a member of All-China Youth Federation. He received the IEEE Marconi Prize Paper Award in 2009 and the IEEE ComSoc Asia Pacific Board Best Young Researcher Award in 2011. He was a recipient of the National May First Labor Medal and the China Youth May Fourth Medal. He has served as the TPC Co-Chair for the IEEE VTC-Spring in 2011 and the Symposium Co-Chair for the IEEE ICC and GLOBECOM. He serves as an Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS.

Anthony Ephremides (Life Fellow, IEEE) received the Ph.D. degree in electrical engineering from Princeton University in 1971.

He holds the Cynthia H. Kim Professorship of information technology with the Department of Electrical and Computer Engineering, University of Maryland, College Park, where he is a Distinguished University Professor and has a joint appointment with the Institute for Systems Research, of which he was among founding members in 1986. He has been with the University of Maryland since 1971. He is the author of several hundred articles and conference presentations. He also holds patents. His research interests lie in the areas of communication systems and networks and all related disciplines such as information theory, control and optimization, satellite systems, queuing models, and signal processing. He is especially interested in wireless networks, energy efficient systems, and the new notion of age of information.