Optimal Policies for Age and Distortion in a Discrete-Time Model

Yunus İnan, Reka Inovan, Emre Telatar
EPFL, Lausanne, Switzerland
Email: {yunus.inan, reka.inovan, emre.telatar}@epfl.ch

***Abstract*—We propose a simple model to study the tradeoff between timeliness and distortion, where different pieces of data have a different cost of not being sent. We pose the question of finding the optimal tradeoff as a policy design problem amenable to dynamic programming methods. We study the structural properties of optimal transmission policies, give an algorithmic procedure to find the optimal tradeoff, and numerically evaluate some instances.**

***Index Terms*—Age of Information, Distortion, Markov Decision Process, Policy Iteration**

I. Introduction

The timeliness of information is a crucial aspect of communications. Stale data may have highly undesirable effects; think, for example, of sensor output for self-driving vehicles, position of an airplane, coolant temperature in a power plant, etc. This aspect of data is nicely captured by the recent studies on Age-of-Information (AoI), by shifting the focus from delay to freshness. At the same time, not all data is equally important. If, in an attempt to reduce staleness our system drops important pieces of data, the remedy may be worse than the disease. In this paper, we study a simple setup where the freshness and importance aspects may be treated together.

The loss, or misrepresentation of data and assigning higher cost to more important data is well captured by the tools of rate distortion theory. As said above, the question of freshness has been an object of study in the AoI literature initiated by Kaul et al. [1]. Since the introduction of AoI, there has been various uses of this metric in many applications. For a comprehensive survey of works in this area, see [2].

However, the combination of importance and freshness has been comparatively less widely studied. Going beyond the AoI metric, in [3] a problem of generating timely updates in a remote estimation setting has been proposed. The authors have investigated the MSE-optimal and AoI-optimal strategies for the estimation of a Wiener process through a queue, and concluded that they are different. In [4], the authors generalized the settings to include Ornstein-Uhlenbeck process. Apart from this approach, there also has been several works on integrating the notion of different data importance and timeliness, e.g., by introducing non-linear cost to stale data [5], [6], by considering separate data streams with different priority [7], [8], or by incorporating the notion of data value which decays with age and selective encoding [9], [10].

In this work, we quantify the notion of importance by using a distortion metric. We propose a simple model which allows us to analyze the tradeoff between timeliness as measured by AoI and the distortion of the data. The tradeoff can be studied by casting it as a Markov Decision Process (MDP). We show that the optimal policy for this MDP can be achieved by a system with finite memory and we also provide an explicit algorithm to compute this policy.

II. Problem Definition

Consider a model composed of the data to be sent, the sender-receiver pair, and the channel in between.

We model the data as an independent and identically distributed (i.i.d.) sequence of random variables $(X_t)_{t\in\mathbb{N}}$. We restrict X to be discrete, taking values in a finite set $\mathcal{X}$. The sender observes X_t at time t and keeps X_t in its buffer.

The channel is modeled as follows: The sender is allowed to speak at times $T_1, T_2, \ldots$. The process $(T_i)_{i\in\mathbb{N}}$ is independent of the process $(X_t)_{t\in\mathbb{N}}$. We further assume that the interspeaking times $Z_i = T_i - T_{i-1}$ are discrete, i.i.d., strictly positive, and with $E[Z^2] < \infty$, e.g., a Geometric random variable with $\Pr(Z_i = t) = p(1-p)^{t-1}$ for $t \geq 1$. The model is inspired by MAC layer protocols giving each sender a turn to speak. At each speaking time T_i, the sender chooses an index $S_i \leq T_i$ and sends X_{S_i} in a packet whose header identifies the timestamp S_i. We require that $S_i < S_{i+1}$, i.e., once a particular X_{S_i} is sent, no data from further past can be conveyed at a later time.

We assume noiseless and zero-delay transmissions between the sender and the receiver. At time t, the receiver has observed X_{S_i} for every i such that $T_i < t$. Consequently at time t, it can reconstruct the data as $Y_j(t) = X_j$ or $Y_j(t) = ?$ depending if X_j is among the receiver's observation up to time t or not.

Now we introduce distortion and timeliness to our model. Given a distortion metric $d : \mathcal{X} \times \mathcal{X} \cup \{?\} \to \mathbb{R}_{\geq 0}$, with

$$d(x,x) = 0 \text{ and } d(x,?) = v(x)$$

define

$$D_t := \frac{1}{t}\sum_{i=1}^{t} d(X_i, Y_i(t)) \text{ and } D := E\left[\limsup_{t\to\infty} D_t\right];$$

and with $i(t) := \sup\{i \geq 0 : T_i < t\}$, $T_0, S_0 = 0$; define for all t,

$$\Delta_t := t - S_{i(t)-1}, \text{ and } \Delta := E\left[\limsup_{t\to\infty} \frac{1}{t}\sum_{i=1}^{t} \Delta_t\right]. \quad (1)$$

Observe that $Y_i(t)$ is equal to X_i or to '?', so the distortion metric defined as above, specifying only $d(x,x)$ and $d(x,?)$ is sufficient to evaluate D. At time $t = T_i$, the transmitter needs to choose S_i on the basis of X_1^t. Since the transmitter's strategy is evaluated by (Δ, D), the transmitter may as well base its choice on V_1^t with $V_t = v(X_t)$. Intuitively, V_i represents an importance score for the packet i; high V_i means the content has high importance and not sending it incurs a high penalty.

Observe that the structure of the problem stays the same it all the elements in $\mathcal{V}$ are multiplied by a positive constant. If $\mathcal{V}$ does not contain 0, then without loss of generality one can assume that the minimum element in $\mathcal{V}$ is 1 and it is an ordered set as $1 = v_1 < v_2 < \ldots < v_{|\mathcal{V}|} := v_{\max} < \infty$. *For the rest of the work, we assume* $|\mathcal{V}| = 2$.

Given the description of the model, the main question now is: What are the achievable (Δ, D) pairs? We attempt to answer this question in the next section and conclude this section with a few remarks.

(i) The model we propose resembles a remote estimation problem of a discrete-time stochastic process through a discrete-time queue. However, we require that the sender sends a packet exactly at speaking times, which is equivalent to force the sender to send as soon as the queue is idle in a discrete-time queueing setting. In [11] and [3], it is shown that the optimal policies need not be of this type. This makes our problem different and allows us to make the relaxation that S_i need not to be stopping times.

(ii) A more sophisticated receiver could try to reconstruct the missing X_t's from the X_{S_i} it has observed. This may be possible even when the process $(X_t)_{t\in\mathbb{N}}$ is i.i.d., if the transmitter chooses S_i appropriately; e.g., by favoring S_i's for which $X_{S_i} = X_{S_i-1}$. The formulation we have adopted does not take into account such receivers.

(iii) If $0 \in \mathcal{V}$, there are multiple interpretations. $V_t = 0$ can be interpreted as either the data is totally trivial (need not be reconstructed), or interpreted as the source having not generated data at time t. The second interpretation allows us to model a source which generates data at intermittent times. Now there is the question of allowing X_t to be sent or not. Our model allows sending of X_t, i.e., in the second interpretation, informs the receiver that there has not been any data generated by the source, and Δ_t decreases accordingly. The reduction of Δ_t can be avoided by appropriate reformulation, see Remark 1. Thus for the rest of the paper, assume $v_{\min}$ is either 0 or 1.

III. MAIN RESULTS

A. Dynamic Programming Formulation

Note that at time T_i and given $(S_i, V_1^{T_i})$, $V_1^{T_{i+1}}$ is independent of the past. This follows from the fact that the importance values V_t's and the interarrival times Z_i are i.i.d. and also independent of each other. Also observe that once S_i is chosen, no future packet can contain data with index less than S_i. Hence, the only relevant information at time T_i is

$$\boldsymbol{B}_i := V_{S_{i-1}+1}^{T_i}.$$

In general, the initial buffer state might result in different (Δ, D) pairs even though the sequence of selected indices $(S_n)_{n\in\mathbb{N}}$ remains same. We assume that the buffer is empty just after $t = 0$. The transmitter is completely specified by the policy $\boldsymbol{B}_i \mapsto S_i$, identifying the data to be transmitted. To obtain the boundary of feasible (Δ, D) pairs, we propose the following cost function:

$$J_i(\eta)^{(S_1^{T_i})} := E\left[\sum_{j=1}^{i} \frac{1}{\mu} D(\boldsymbol{B}_j, S_{j-1}, S_j) + \eta(T_j - S_j)\right], \quad \eta > 0$$

with $\mu = E[Z]$ and

$$D(\boldsymbol{b}, s, s') := \sum_{k=s+1}^{s'-1} b_k.$$

We also define

$$J(\eta)^{(\boldsymbol{S})} := \limsup_{i\to\infty} \frac{1}{i} J_i(\eta)^{(S_1^{T_i})}.$$

where $\boldsymbol{S} := (S_n)_{n\in\mathbb{N}}$. We seek to minimize $J(\eta)^{(\boldsymbol{S})}$ over the policies of choosing $\boldsymbol{S}$, formulated as the following optimization problem:

$$J^*(\eta) := \inf_{\boldsymbol{S}} J(\eta)^{(\boldsymbol{S})} \tag{2}$$

where the infimum is over all policies that map the buffer content to the index of the transmitted data. We now formulate (2) as a MDP optimization and relate Δ and D to $J^*(\eta)$.

We note that

$$\limsup_{t\to\infty} \frac{1}{t} \sum_{t'=1}^{t} \Delta_{t'} = \limsup_{i\to\infty} \frac{\sum_{j=1}^{i} Q_j}{\sum_{j=1}^{i} Z_j}$$

where

$$Q_j := (T_j - S_j) Z_{j+1} + \frac{Z_{j+1}(Z_{j+1}+1)}{2}.$$

Since $\lim_{i\to\infty} \frac{1}{i} \sum_{j=1}^{i} \frac{Z_{j+1}(Z_{j+1}+1)}{2} = \frac{1}{2} E[Z(Z+1)] =: \nu$ and $\lim_{i\to\infty} \frac{1}{i} \sum_{j=1}^{i} Z_j = \mu$ with probability 1 by Law of Large Numbers, we obtain

$$\Delta = E\left[\frac{1}{\mu} \limsup_{i\to\infty} \frac{1}{i} \sum_{j=1}^{i} (T_j - S_j) Z_{j+1} + \frac{\nu}{\mu}\right].$$

Note that the Δ cannot be smaller than ν/μ. We subtract ν/μ to obtain the excess age, given by

$$\Delta_e := E\left[\frac{1}{\mu} \limsup_{i\to\infty} \frac{1}{i} \sum_{j=1}^{i} (T_j - S_j) Z_{j+1}\right]$$

and determine the feasible (Δ_e, D) pairs.

Remark 1. *To cover the case that $v = 0$ is interpreted as the source having not generated any data and Δ_t should not decrease upon the sending of $v = 0$; one can replace Z_i with $\tilde{Z}_i$, which represents the first interspeaking time after the source generates some data, and proceed similarly.*

To be able to write simple expressions for Δ_e, D and relate those to one-step costs of a dynamic programming problem, we make a technical assumption that $\sup_i E[(T_i - S_i)^2] < \infty$. We give these simple expressions in the theorem below. The proof is found in Appendix A.

Theorem 1. *For policies with* $\sup_i E[(T_i - S_i)^2] < \infty$,

$$\Delta_e = E\left[\limsup_{i\to\infty} \frac{1}{i} \sum_{j=1}^{i} (T_j - S_j)\right],$$

$$D = \frac{1}{\mu} E\left[\limsup_{i\to\infty} \frac{1}{i} \sum_{j=1}^{i} D(\boldsymbol{B}_j, S_{j-1}, S_j)\right].$$

Furthermore, $J(\eta)^{(\boldsymbol{S})} \leq D^{(\boldsymbol{S})} + \eta\Delta_e^{(\boldsymbol{S})}$ and for stationary policies the equality holds. Hence, solving the optimization problem in (2) *gives a lower bound in general.*

Given the current buffer content $\boldsymbol{B}_j$, the next state depends only on $\boldsymbol{B}_j$ and S_j. Hence, the optimization problem (2) can be formulated as dynamic programming where the states of subject MDP are described by the buffer content $\boldsymbol{B}_j$. More precisely, our formulation is an infinite-horizon average-cost dynamic programming problem with states $\boldsymbol{b} \in \mathcal{V}^*$ in MDP terminology [12]. For analyzing this dynamic programming problem, let us recall:

Definition 1 (Unichain policy, [12])**.** *If a deterministic stationary policy $s(\boldsymbol{b})$ induces a Markov Chain with a single recurrent class and a possibly empty set of transient states, it is called unichain.*

In an average-cost dynamic programming setting, we evaluate a unichain policy $s(\boldsymbol{b})$, $\boldsymbol{b} \in \mathcal{V}^*$ by solving the linear system with unknowns $h(\boldsymbol{b})$, $\boldsymbol{b} \in \mathcal{V}^*$ and λ; given by

$$h(\boldsymbol{b}) + \lambda = \frac{1}{\mu} \sum_{k=1}^{s(\boldsymbol{b})-1} b_k + \eta(l(\boldsymbol{b}) - s(\boldsymbol{b})) + E[h(\boldsymbol{b}^{l(\boldsymbol{b})}_{s(\boldsymbol{b})+1}, \boldsymbol{V}^Z)], \tag{3}$$

where $l(\boldsymbol{b})$ is the length of $\boldsymbol{b}$, i.e. the current buffer length, Z is the next interspeaking time, $\boldsymbol{V}^Z$ is a vector of i.i.d. V's of length Z, $h(\boldsymbol{b})$ is called the relative value of state $\boldsymbol{b}$, and λ is the average cost induced by this policy. Since (3) determines $h(\boldsymbol{b})$ only up to an additive constant, we take a reference state (see [12, Chapter 4]) as one of the $\boldsymbol{b} \in \mathcal{V}^*$ and we set $h(\boldsymbol{b}) = 0$. We also note that for a unichain policy, the linear system given by (3) has a unique solution [12].

B. Policy Iteration with a Truncated State Space

The MDP given in the previous subsection has a countably infinite state space, hence optimal policies are very hard to analyze. Furthermore, the existence of a stationary policy achieving the infimum in (2) is not guaranteed in general. This leads us to consider a finite state version of the problem by limiting the buffer size to K. The number of states will be then finite. Denote this state space by $\mathcal{V}^{\leq K} := \cup_{l \leq K} \mathcal{V}^l$. Surprisingly, as we shall see in III-C, the optimal policy of the infinite state space problem will base its decisions only on a bounded buffer. Thus, the restriction we make here is provided that K is large enough.

Also observe that in this case the expectation on the RHS of (3) involves terms $h(\tilde{\boldsymbol{b}})$ with $l(\tilde{\boldsymbol{b}})$ which might be more than K. Such terms should be evaluated as $h(\tilde{\boldsymbol{b}}) = \sum_{k=1}^{l(\tilde{\boldsymbol{b}})-K} \tilde{b}_k + h(\tilde{\boldsymbol{b}}^{l(\tilde{\boldsymbol{b}})}_{l(\tilde{\boldsymbol{b}})-K+1})$.

Let $p_i := \Pr(Z = i)$ and $q_i := \Pr(Z > i)$. With a slight modification of (3), we can now evaluate a unichain stationary policy $s(\boldsymbol{b})$, $\boldsymbol{b} \in \mathcal{V}^{\leq K}$ by solving the linear system

$$\begin{aligned} h(\boldsymbol{b}) + \lambda = {} & \frac{1}{\mu} \sum_{k=1}^{s(\boldsymbol{b})-1} b_k + \eta(l(\boldsymbol{b}) - s(\boldsymbol{b})) \\ & + \frac{1}{\mu} \sum_{k=s(\boldsymbol{b})+1}^{l} b_k q_{K+k-l-1} + \frac{E[V]}{\mu} E[(Z-K)^+] \\ & + \sum_{k=1}^{K-l+s(\boldsymbol{b})} p_k E[h(\boldsymbol{b}^l_{s(\boldsymbol{b})+1}, \boldsymbol{V}^k)] \\ & + \sum_{k=K-l+s(\boldsymbol{b})+1}^{K-1} p_k E[h(\boldsymbol{b}^l_{k-(K-l)+1}, \boldsymbol{V}^k)] \\ & + q_{K-1} E[h(\boldsymbol{V}^K)], \end{aligned} \tag{4}$$

with $h(b) = 0$ for $b \in \mathcal{V}$ if $p_1 > 0$. Otherwise, one can choose another reference state of greater length, e.g., $(v_{\min}, v_{\min})$. We give a brief description of the policy iteration algorithm [12]:

Algorithm 1: Policy iteration

1. Start with stationary policy $s^{(0)}(\boldsymbol{b}) = l(\boldsymbol{b})$.
2. Evaluate $s^{(i)}(\boldsymbol{b})$ according to (4) to find $h^{(i)}(\boldsymbol{b})$, $\boldsymbol{b} \in \mathcal{V}^{\leq K}$ and $\lambda^{(i)}$.
3. For all $\boldsymbol{b} \in \mathcal{V}^{\leq K}$ set $s^{(i+1)}(\boldsymbol{b}) = \arg\min_{s \leq l(\boldsymbol{b})}$ RHS of (4).
4. If $s^{(i+1)}(\boldsymbol{b}) = s^{(i)}(\boldsymbol{b})$ for all $\boldsymbol{b} \in \mathcal{V}^{\leq K}$, terminate. Else go to step 2.

Remark 2. *Intuitively, step 3 of the above algorithm modifies $s^{(i)}(\boldsymbol{b})$ in the following way: Consider two processes starting at $\boldsymbol{b}$. The first one is iterated with respect to $s^{(i)}$, whereas the second one is iterated with an $\tilde{s}(\boldsymbol{b})$ at the first step and the following steps will be based on $s^{(i)}$. Now consider the expected accumulated costs of these two processes until they reach the same state. If the second process has less expected accumulated cost, change all $s^{(i)}(\boldsymbol{b})$ to $\tilde{s}(\boldsymbol{b}) = s^{(i+1)}(\boldsymbol{b})$ to obtain a better policy. If perturbing at the first step does not improve the cost, then $s^{(i+1)}(\boldsymbol{b}) = s^{(i)}(\boldsymbol{b})$.*

Although the number of states is now finite, it is not clear that the policy iteration algorithm yields a unichain policy. We shall prove this in the following lemma.

Lemma 1. *If the buffer size is limited to K, the policy iteration terminates with a unichain policy.*

Proof: We use the fact that if $s^{(i)}(\boldsymbol{b}) \neq l(\boldsymbol{b})$, then $v_{s^{(i)}(\boldsymbol{b})} \neq v_{\min}$ for $i \geq 1$, i.e., never take $v_{\min}$ if it is not at the end. This is proven by observing that $s^{(i)}(\boldsymbol{b})$ can never minimize the RHS of (4) otherwise. Now take any $\boldsymbol{s}^{(i)}$, $i \geq 1$. Eventually, at some point the buffer content will consist of only $v_{\min}$s and according to $\boldsymbol{s}^{(i)}$, the sender sends the most recently arrived data. At this moment, the buffer will be renewed and this shows that there can only be one recurrent class. Hence, any $\boldsymbol{s}^{(i)}$ for $i \geq 1$ is unichain and the policy iteration terminates with a unichain policy. ∎

Lemma 1 tells that the policy iteration terminates with a unichain policy. Since the optimal policy cannot choose $v_{\min}$ unless it is at the end, (see Property 1 (iv) below) this policy will be optimal for the case with limited buffer size [12].

C. The Exact Buffer Size Needed for Optimal Solution

Truncating the state space restricts the actions that may be taken. Therefore, in general, the infimum in (2) may not

be attained with a truncated buffer. As we have said in the previous section, for our problem this is not the case and the infimum is indeed attained with a finite buffer size. In this section we quantify this buffer size.

First, consider the policy $s(\boldsymbol{b}) = l(\boldsymbol{b})$ for all $\boldsymbol{b} \in \mathcal{V}^{\leq K}$, i.e., always send the most recent element in the buffer. One can observe that this policy induces a Markov Chain with only $|\mathcal{V}| = 2$ states regardless of K. We shall now show that this policy is optimal for η above some threshold $\eta_{\max}$.

Lemma 2. *For $\eta \geq \eta_{\max} := \frac{1}{\mu}(v_{\max} - v_{\min})$ and for any $M \geq 1$, the optimal policy among $\mathcal{V}^{\leq M}$ is $s(\boldsymbol{b}) = l(\boldsymbol{b})$; which can be implemented with a buffer size of 1.*

Proof: We show that the policy $s(\boldsymbol{b}) = l(\boldsymbol{b})$ will be unchanged by the policy iteration. Start the policy iteration with $s^{(0)}(\boldsymbol{b}) = l(\boldsymbol{b})$. Recalling Remark 2, we observe that perturbing the policy at initial step cannot decrease the expected accumulated cost until the original and perturbed processes coincide. More precisely, the two processes will coincide immediately at the next step and the difference of the accumulated costs will be $k\eta - \frac{1}{\mu}(b_{l(\boldsymbol{b})-k} - b_{l(\boldsymbol{b})}) \geq \eta - \frac{1}{\mu}(v_{\max} - v_{\min}) \geq 0$. Hence the policy remains unchanged. ∎

Now considering the state space $\mathcal{V}^{\leq K}$, we give some properties of optimal policies whose derivations are given in Appendix B.

Property 1. *For an optimal stationary policy $s^*(\boldsymbol{b})$, and optimal relative values $h^*(\boldsymbol{b})$,*

(i) *For any state $(\boldsymbol{b}, \boldsymbol{b}')$, $s^*(\boldsymbol{b}, \boldsymbol{b}') = l(\boldsymbol{b}) + s^*(\boldsymbol{b}')$ or $s^*(\boldsymbol{b}, \boldsymbol{b}') \leq l(\boldsymbol{b})$*
(ii) *$h^*(\boldsymbol{b}') \leq h^*(\boldsymbol{b})$ for $\boldsymbol{b}'$, $\boldsymbol{b} \in \mathcal{V}^l$ if $b'_i \leq b_i$ for all $i \leq l$*
(iii) *$h^*(\boldsymbol{b}') \leq h^*(\boldsymbol{b}, \boldsymbol{b}') \leq \frac{1}{\mu}(b_1 + \ldots + b_{l(\boldsymbol{b})}) + h^*(\boldsymbol{b}')$ for $\boldsymbol{b}$, $\boldsymbol{b}' \in \mathcal{V}^{\leq K}$ such that $l(\boldsymbol{b}) + l(\boldsymbol{b}') \leq K$*
(iv) *If $s^*(\boldsymbol{b}) \neq l(\boldsymbol{b})$, then $v_{s^*(\boldsymbol{b})} \neq v_{\min}$*

Corollary 1. *For the state $\boldsymbol{b} = (v_{\max}, v_{\min}^{L-1})$;*

$$s^*(\boldsymbol{b}) = \begin{cases} 1, & \eta \leq \frac{1}{\mu}\frac{(v_{\max} - v_{\min})}{L-1} \\ L, & \eta > \frac{1}{\mu}\frac{(v_{\max} - v_{\min})}{L-1} \end{cases}. \tag{5}$$

Proof: We know that any policy given at step $i \geq 1$ of the policy iteration algorithm should not choose $v_{\min}$ if it is not at the end. Therefore, the two possible strategies are: choose $v_{\max}$ in the head or $v_{\min}$ in the tail. Referring to Remark 2, suppose $s^{(i)}(\boldsymbol{b}) = l(\boldsymbol{b})$ and $\tilde{s}(\boldsymbol{b}) = 1$. Observe that the original and perturbed processes will coincide immediately after the first iteration and the difference of costs will be $\eta(L-1) + \frac{1}{\mu}(v_{\min} - v_{\max})$. Then, $s^{(i+1)}(\boldsymbol{b}) = 1$ if $\eta(L-1) \leq \frac{1}{\mu}(v_{\max} - v_{\min})$; otherwise $s^{(i+1)}(\boldsymbol{b}) = l(\boldsymbol{b})$. Note that the difference does not depend on i and hence the statement for $\boldsymbol{s}^{(i+1)}$ is also true for $\boldsymbol{s}^*$. ∎

The above corollary therefore gives a necessary buffer size for optimality as it tells that at $\eta = \frac{1}{\mu}\frac{(v_{\max} - v_{\min})}{L-1}$, the first packet in the state $\boldsymbol{b} = (v_{\max}, v_{\min}^{L-1})$ is chosen by the optimal policy. Hence, attaining the optimal policy requires a buffer size of at least $\lceil \frac{1}{\mu}\frac{(v_{\max} - v_{\min})}{\eta} \rceil$. However, this does not imply that the optimal policy is reached within a finite buffer size.

Theorem 2. *For $M \geq K(\eta) := \lceil \frac{1}{\mu}\frac{(v_{\max} - v_{\min})}{\eta} \rceil$, the optimal policy among $\mathcal{V}^{\leq M}$ is attained by a policy with buffer size K.*

Proof Sketch: We use a proof strategy as in Lemma 2. We claim that the policy iteration maps policies with buffer size at most $K(\eta)$ to again policies with buffer size at most $K(\eta)$ (See appendix C for the full proof). Since the policy iteration algorithm converges to a unique solution, we conclude this solution must have a buffer size of at most $K(\eta)$. The assumption that $|\mathcal{V}| = 2$ is essential for our proof. ∎

Corollary 2. *The optimal policy among untruncated state space policies is attained with a buffer size $K(\eta)$.*

Proof Sketch: We note that when the policy iteration terminates the policy it produces not only solves the Bellman equation for the state space $\mathcal{V}^{\leq M}$, for every $M \geq K(\eta)$, but it also solves the Bellman Equation for the state space $\mathcal{V}^*$. Since our problem fulfils the conditions 10.1-10.4' in [13], we conclude that this policy attains the infimum in (2). ∎

D. An Algorithm to Find the (Δ_e, D) Boundary Curve

It is time to combine the results from the previous sections to provide an extension of policy iteration algorithm to find the tangent lines to the feasible (Δ_e, D) region.

Algorithm 2: Extended Policy Iteration Algorithm

```
Choose η_min, ε > 0. Start with the state space S = V,
  η = η_max and the policy s^(0)(b) = l(b)
while η > η_min do
    Find λ^(i), h^(i)(b), b ∈ S by solving (4)
    foreach b ∈ S do
        s^(i+1)(b) ← arg min_{s≤l(b)} RHS of (4)
    if s^(i+1)(b) = 1 then
        foreach v ∈ V do
            S ← S ∪ (b_1, v, b_2^{l(b)})
    if s^(i+1)(b) = s^(i)(b) for all b ∈ S then
        J*(η) ← λ^(i)
        η ← η − ε.
return all (η, J*(η)) pairs
```

One could also define the state space as $\mathcal{V}^{\leq K}$ in the beginning of Algorithm 2. Although the complexity stays asymptotically same in that case, starting with a smaller state space and gradually expanding is pratically more efficient in terms of time and memory.

Theorem 3. *The algorithm finds the optimal curve*

$$D(\Delta_e) = \sup_{\eta > 0} J^*(\eta) - \eta\Delta_e. \tag{6}$$

Proof: Since at most 2^{K+1} states are added the policy iteration converges in a finite time, and the algorithm yields an approximation to $J^*(\eta)$. Since the state space is bounded, the condition $\sup_j E[(T_j - S_j)^2] < \infty$ from Theorem 1 is satisfied. Furthermore, we obtain a stationary policy, implying that $J^*(\eta) = D + \eta\Delta_e$. Finally, one takes the convex conjugate $D(\Delta_e) = \sup_{\eta > 0} J^*(\eta) - \eta\Delta_e$ to obtain the boundary curve

for the feasible (Δ_e, D) pairs. By choosing $\eta_{\min}$ and ϵ small enough, the curve can be approximated arbitrarily closely. ■

However, the necessary buffer size scales with $\frac{1}{\eta}$ and the algorithm may not be efficient. This suggests that even though the algorithm gives the almost exact curve, it is impractical to do so. To overcome this difficulty, one may rely on approximate dynamic programming algorithms; or resort to Monte Carlo estimations for the policy evaluation [12].

We have made some computations with Algorithm 2, and we have not observed any simple structure for optimal policies. We also evaluated simple policies given below and compared their performance with the curve generated by Algorithm 2, referred as (DP) in figures below. These simple policies are:

(S1) Send oldest important data with a maximum buffer size K.
(S2) Send the newest important data with a maximum buffer size K.
(S3) Send the newest important data that has arrived more than K slots ago.

To compare these strategies, we also give a simple converse bound and observe their approach towards this bound for large Δ_e.

Lemma 3. *Suppose $V = v_{\max}$ with probability α. Then for any Δ_e, $D > D_{\min} = \left(1 - \alpha + \min\left\{\alpha - \frac{1}{\mu}, 0\right\}\right)v_{\min} + \max\left\{\alpha - \frac{1}{\mu}, 0\right\}v_{\max}$.*

Proof Sketch: The sender can send at most $\frac{1}{\mu}$ fraction of the data. We then optimize over its selection of data to obtain the result. ■

Note that another converse bound is produced by Algorithm 2 for any $\eta_{\min}$ bounded away from zero as the boundary curve is given by (6), and the supremum is taken over $\eta > \eta_{\min}$. Hence we can choose $\eta_{\min}$ such that for a maximum buffer size of $\left\lceil \frac{v_{\max} - v_{\min}}{\mu \eta_{\min}} \right\rceil$, the algorithm terminates in a timely manner. This straight line converse bound will be referred as (DP Converse) in figures below.

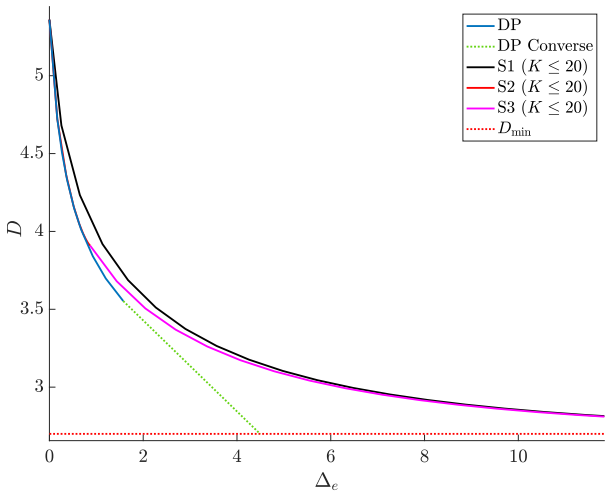

Fig. 1. Comparison of the strategies for $\mathcal{V} = \{1, 20\}$ and $\Pr(V = 1) = 0.7$. Z is taken as a Geometric random variable with success probability 0.2. (S2 is almost entirely hidden by DP)

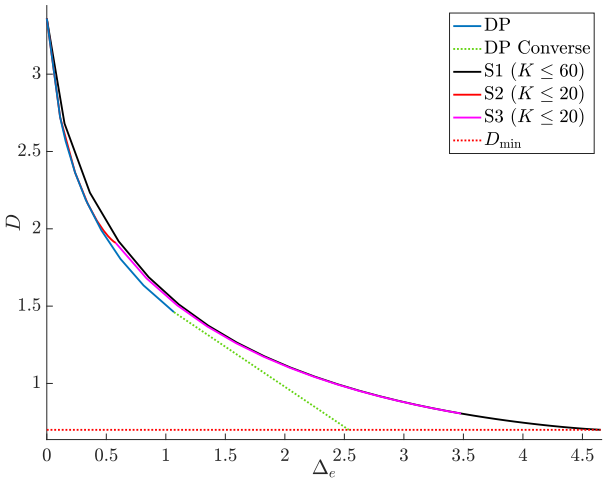

Fig. 2. Comparison of the strategies for $\mathcal{V} = \{1, 20\}$ and $\Pr(V = 1) = 0.8$. Z is taken as a Geometric random variable with success probability 0.3.

IV. Discussion

In the absence of a distortion measure, it is clear that the optimal strategy is to send the freshest data in the buffer at each speaking time T_i as this will minimize the age. However, if this freshest data is an unimportant packet, it may be beneficial to send an important packet instead, sacrificing freshness for lowering distortion. Naively, this would argue that if the freshest data is not to be sent, one should send the freshest important packet. Some thought reveals that this is not optimal: such a strategy lowers the chance of finding an important data in the future buffer. Having observed this, we tried to prove the optimality of several easy-to-describe policies. These attempts were not successful, and indeed, the optimal policies found by policy iteration methods do not seem easy-to-describe. Despite this, it was not a priori clear to us that the optimal policies would turn out to be of bounded buffer size, it was a surprise that this was the case for $|\mathcal{V}| = 2$. However, computations support the claim that a bounded buffer suffices for $|\mathcal{V}| > 2$ as well, i.e., Algorithm 2 terminates.

Note that the optimal policy may differ with different interspeaking time distributions. One could study which distribution is best (or worst) among those with a given mean.

Further note that as η (the weight of freshness) gets small, the required buffer size increases and the search space for policies becomes too large to find the optimal policy. The absence of a good converse bound makes it hard to know how far from the optimal a given policy is. The converse bounds shown on the plots are not strong enough for such a purpose.

The method described in Section III, with minor modifications, allows finding optimal policies for Markov $(X_t)_{t \in \mathbb{N}}$. However, appropriateness of the distortion metric we have adopted is doubtful, as our remark (ii) at the end of Section II applies even more strongly in this scenario.

One can change the transmission model to go beyond sending packets that contain a single X_i. E.g., at each speaking time we are allowed to send L bits; this should convey not only data but also information that identifies the time index (or indices) the data pertains to. This kind of 'coded' transmission in which we can arbitrarily map the past X_i's to the current transmission would give another degree of freedom to improve performance. Whether the improvement justifies the additional complexity is something that seems worth studying.

REFERENCES

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

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

[3] Y. Sun, Y. Polyanskiy, and E. Uysal-Biyikoglu, "Remote estimation of the wiener process over a channel with random delay," in *2017 IEEE International Symposium on Information Theory (ISIT)*, 2017, pp. 321–325.

[4] T. Z. Ornee and Y. Sun, "Sampling for remote estimation through queues: Age of information and beyond," in *2019 International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT)*, 2019, pp. 1–8.

[5] A. Kosta, N. Pappas, A. Ephremides, and V. Angelakis, "Age and value of information: Non-linear age case," in *2017 IEEE International Symposium on Information Theory (ISIT)*, 2017, pp. 326–330.

[6] ——, "The cost of delay in status updates and their value: Non-linear ageing," *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 4905–4918, 2020.

[7] S. K. Kaul and R. D. Yates, "Age of information: Updates with priority," in *2018 IEEE International Symposium on Information Theory (ISIT)*, 2018, pp. 2644–2648.

[8] E. Najm, R. Nasser, and E. Telatar, "Content based status updates," *IEEE Transactions on Information Theory*, vol. 66, no. 6, pp. 3846–3863, 2020.

[9] B. Buyukates, M. Bastopcu, and S. Ulukus, "Optimal selective encoding for timely updates with empty symbol," in *2020 IEEE International Symposium on Information Theory (ISIT)*, 2020, pp. 1794–1799.

[10] M. Bastopcu and S. Ulukus, "Age of information for updates with distortion," in *2019 IEEE Information Theory Workshop (ITW)*, 2019, pp. 1–5.

[11] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff, "Update or wait: How to keep your data fresh," *IEEE Transactions on Information Theory*, vol. 63, no. 11, pp. 7492–7508, 2017.

[12] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Vol. II*, 3rd ed. Athena Scientific, 2007.

[13] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, 1st ed. USA: John Wiley & Sons, Inc., 1994.

[14] D. Williams, *Probability with Martingales*, ser. Cambridge mathematical textbooks. Cambridge University Press, 1991.

[15] S. Resnick, *A Probability Path*, ser. Modern Birkhäuser Classics. Birkhäuser Boston, 2003. [Online]. Available: https://books.google.ch/books?id=T9-PMDSVDNsC

APPENDIX

A. Proof of Theorem 1

Define $W_j := T_j - S_j$ for the rest of the proof. First, we use Lemmas 4,5,6 to derive the simple expressions. Then, we prove the inequality $J(\eta)^{(\boldsymbol{S})} \leq D^{(\boldsymbol{S})} + \eta\Delta_e^{(\boldsymbol{S})}$. We denote 'almost surely' by *a.s.* The proof is also valid for $|\mathcal{V}| > 2$.

Lemma 4. $\frac{1}{i}\sum_{j=1}^{i} W_j(Z_{j+1} - \mu) \to 0$ *a.s. if* $\sum_j \frac{E[W_j^2]}{j^2} < \infty$.

Proof: We will follow similar steps to those in the proof of Strong Law of Large Numbers given in Williams' book [14]. We use the result that if $\sum b_i/i$ converges, then $\frac{1}{i}\sum_{j\leq i} b_j \to 0$. Therefore, it is sufficient to check if $\sum_i W_i(Z_{i+1} - \mu)/i$ converges *a.s.* We now show that $M_n := \sum_{i=2}^{n} W_{i-1}(Z_i - \mu)/(i-1)$, $M_1 := 0$ is a martingale with respect to the filtration $\mathcal{F}_n := \sigma(Z_1, \ldots, Z_n, \boldsymbol{X}_1^{T_n})$.

Observe that $E[M_n|\mathcal{F}_{n-1}] = M_{n-1} + E[W_{n-1}(Z_n - \mu)|\mathcal{F}_{n-1}]/(n-1) = M_{n-1}$ as W_{n-1} is $\mathcal{F}_{n-1}$-measurable and Z_n is independent of $\mathcal{F}_{n-1}$ with $E[Z_n] = \mu$. Since M_n consists of uncorrelated increments, one can write

$$E[M_n^2] = \sum_{i=2}^{n} \frac{E[W_{i-1}^2]\text{Var}(Z)}{(i-1)^2}.$$

Note that we assumed $E[Z^2] < \infty$, hence $\text{Var}(Z) < \infty$. Moreover, if $\sum_i \frac{E[W_i^2]}{i^2} < \infty$, then $\sup E[M_n^2] < \infty$, and by Martingale Convergence Theorem, M_n converges *a.s.* ∎

Lemma 5. *For policies satisfying* $\sup_j E[W_j^2] < \infty$,

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{t'=1}^{t} \Delta_{t'} = \limsup_{i\to\infty} \frac{1}{i}\sum_{j=1}^{i} W_j.$$

Proof: Since $\sup_j E[W_j^2] < \infty$, $\frac{1}{i}\sum_{j=1}^{i} W_j Z_{j+1} \to \frac{1}{i}\sum_{j=1}^{i} W_j\mu$ *a.s.* by Lemma 4. Hence,

$$\limsup_{i\to\infty} \frac{1}{i}\sum_{j\leq i} W_j Z_{j+1} = \mu \limsup_{i\to\infty} \frac{1}{i}\sum_{j\leq i} W_j$$

and the proof is done. ∎

Similar to the derivation of Δ, for policies satisfying $\sup_j E[W_j^2] < \infty$ we show that D can also be written in a simple form.

Lemma 6. *For policies satisfying* $\sup_j E[W_j^2] < \infty$,

$$\limsup_{t\to\infty} D_t = \frac{1}{\mu}\limsup_{i} \frac{1}{i}\sum_{j=1}^{i} D(\boldsymbol{B}_j, S_j, S_{j-1}) \tag{7}$$

Proof: Define $i = i(t) := \sup\{i \geq 0 : T_i \leq t\}$ and write D_t as

$$D_t = \frac{1}{t}\sum_{j\leq S_i} d(X_j, Y_j) + \frac{1}{t}\sum_{j=S_i+1}^{t} d(X_j, Y_j(t))$$

as all the estimates until S_i are finalized. Upper bound D_t as

$$\begin{aligned} D_t &\leq \frac{1}{t}\sum_{j\leq S_i} d(X_j, Y_j) + \frac{1}{t}(T_{i+1} - S_i - 1)v_{\max} \\ &\leq \frac{1}{T_i}\sum_{j\leq S_i} d(X_j, Y_j) + \frac{1}{t}(Z_{i+1} + W_i - 1)v_{\max}. \end{aligned}$$

Since $\sup_j E[W_j^2] < \infty$, W_i is *a.s.* finite for all j and hence $Z_{j+1} + W_j$ is *a.s.* finite. Thus $\frac{1}{t}(Z_{j+1} + W_j - 1)v_{\max} \to 0$ *a.s.* Then, we obtain

$$\limsup D_t \leq \limsup_i \frac{1}{T_i}\sum_{j\leq S_i} d(X_j, Y_j).$$

Now, lower bound D_t as

$$\begin{aligned} D_t \geq \frac{1}{t}\sum_{j\leq S_i} d(X_j, Y_j) &\geq \frac{1}{T_{i+1}}\sum_{j\leq S_i} d(X_j, Y_j) \\ &= \frac{1}{T_i + Z_{i+1}}\sum_{j\leq S_i} d(X_j, Y_j) \end{aligned}$$

and obtain

$$\limsup D_t \geq \limsup_i \frac{1}{T_i + Z_{i+1}}\sum_{j\leq S_i} d(X_j, Y_j).$$

Finally, observe that $\frac{T_i}{i} \to \mu$ and $\frac{Z_{i+1}}{i} \to 0$ *a.s.* Hence,

$$\limsup D_t = \frac{1}{\mu}\limsup_i \frac{1}{i}\sum_{j\le S_i} d(X_j, Y_j)$$
$$= \frac{1}{\mu}\limsup_i \frac{1}{i}\sum_{j=1}^{i} D(\boldsymbol{B}_j, S_j, S_{j-1})$$

■

The final assertion of Theorem 1 is proven by noting that since $\sup_j E[W_j^2] < \infty$, it follows that $\sup_i E[(\frac{1}{i}\sum_{j=1}^{i} W_j)^2] < \infty$. Thus, the family $(\frac{1}{i}\sum_{j=1}^{i} W_j)_{i\in\mathbb{N}}$ is uniformly integrable [14, Chapter 13]. One can then use Reverse Fatou's Lemma for uniformly integrable families [15] to obtain

$$\Delta_e = E\left[\limsup_{i\to\infty}\frac{1}{i}\sum_{j=1}^{i} W_j\right] \ge \limsup_{i\to\infty} E\left[\frac{1}{i}\sum_{j=1}^{i} W_j\right].$$

A similar reasoning for D follows from the fact that each $D(\boldsymbol{B}_{T_j}, S_j, S_{j-1})$ is smaller than $v_{\max}(W_{j-1}+Z_j)$ and one proceeds in a similar way to obtain

$$D \ge \limsup_{i\to\infty}\frac{1}{\mu}E\left[\frac{1}{i}\sum_{j=1}^{i} D(\boldsymbol{B}_j, S_j, S_{j-1})\right].$$

As $\limsup_n a_n + \limsup_n b_n \ge \limsup_n (a_n+b_n)$ in general, the inequality

$$J(\eta)^{(\boldsymbol{S})} \le D^{(\boldsymbol{S})} + \eta\Delta_e^{(\boldsymbol{S})}$$

holds in general. If the policy $\boldsymbol{S}$ is stationary, the $\limsup$'s can be changed with $\lim$'s by Renewal theory and the equality holds.

B. Proof of Property 1

In the following proofs, $g(\boldsymbol{b}, s)$ refers to the one step costs and $E_{\boldsymbol{b},s}[h(\boldsymbol{B}')]$ refers to the expected relative value of the next state, i.e., $g(\boldsymbol{b}, s) = \sum_{k=1}^{s-1} b_k + \eta(l(\boldsymbol{b}) - s)$ and $E_{\boldsymbol{b},s}[h(\boldsymbol{B}')] = E[h(b_{s+1}^{l(\boldsymbol{b})}, \boldsymbol{V}^Z)]$

(i) Define $\boldsymbol{q} := (\boldsymbol{b}, \boldsymbol{b}')$. Suppose $s^*(\boldsymbol{q}) \neq l(\boldsymbol{b}) + s^*(\boldsymbol{b}')$ and $s^*(\boldsymbol{q}) > l(\boldsymbol{b})$. Then $s^*(\boldsymbol{q}) = \arg\min_{s\le l(\boldsymbol{q})} g(\boldsymbol{q}, s) + E_{\boldsymbol{q},s}[h(\boldsymbol{B}')] = l(\boldsymbol{b}) + \arg\min_{s\le l(\boldsymbol{b}')} g(\boldsymbol{b}', s) + E_{\boldsymbol{b}',s}[h(\boldsymbol{B}')] = l(\boldsymbol{b}) + s^*(\boldsymbol{b}')$. Hence there is a contradiction.

(ii) During the proof, we use $\boldsymbol{u}$ instead of $\boldsymbol{b}'$ for notational convenience. We want to obtain a lower bound for

$$h^*(\boldsymbol{u}) - h^*(\boldsymbol{b}) = g(\boldsymbol{u}, s^*(\boldsymbol{u})) - g(\boldsymbol{b}, s^*(\boldsymbol{b})) + E_{\boldsymbol{u},s^*(\boldsymbol{u})}[h^*(\boldsymbol{U}')] - E_{\boldsymbol{b},s^*(\boldsymbol{b})}[h^*(\boldsymbol{B}')]$$

To lower bound $E_{\boldsymbol{u},s^*(\boldsymbol{u})}[h^*(\boldsymbol{U}')] - E_{\boldsymbol{b},s^*(\boldsymbol{b})}[h^*(\boldsymbol{B}')]$, consider two MDPs in parallel: One starting from $\boldsymbol{u}$ and the other from $\boldsymbol{b}$. For the process starting from $\boldsymbol{u}$, apply optimal policies and for the process starting from $\boldsymbol{b}$, apply the optimal policies of the first process. Assume they have the same future arrivals for the consequent buffer states. Then they will follow the paths

$$\begin{aligned}\boldsymbol{u} = \boldsymbol{u}^{(0)} &\overset{s^*(\boldsymbol{u}^{(0)})}{\to} \boldsymbol{u}^{(1)} \overset{s^*(\boldsymbol{u}^{(1)})}{\to} \boldsymbol{u}^{(2)} \to \ldots \to \boldsymbol{u}^{(\tau)}\\ \boldsymbol{b} = \boldsymbol{b}^{(0)} &\overset{s^*(\boldsymbol{u}^{(0)})}{\to} \boldsymbol{b}^{(1)} \overset{s^*(\boldsymbol{u}^{(1)})}{\to} \boldsymbol{b}^{(2)} \to \ldots \to \boldsymbol{b}^{(\tau)}\end{aligned} \tag{8}$$

and eventually end in the same state $\boldsymbol{u}^{(\tau)} = \boldsymbol{b}^{(\tau)}$ after a random time τ. This is because the applied policies are same for both processes and eventually the buffers will be occupied by the same arrivals. Observe that

$$\begin{aligned}h^*(\boldsymbol{u}) - h^*(\boldsymbol{b}) &= g(\boldsymbol{u}, s^*(\boldsymbol{u})) - g(\boldsymbol{b}, s^*(\boldsymbol{b}))\\ &\quad + E_{\boldsymbol{u},s^*(\boldsymbol{u})}[h^*(\boldsymbol{U}')] - E_{\boldsymbol{b},s^*(\boldsymbol{b})}[h(\boldsymbol{B}')]\\ &\ge g(\boldsymbol{u}, s^*(\boldsymbol{u})) - g(\boldsymbol{b}, s^*(\boldsymbol{u}))\\ &\quad + E_{\boldsymbol{u},s^*(\boldsymbol{u})}[h^*(\boldsymbol{U}')] - E_{\boldsymbol{b},s^*(\boldsymbol{u})}[h(\boldsymbol{B}')]\\ &= g(\boldsymbol{u}, s^*(\boldsymbol{u})) - g(\boldsymbol{b}, s^*(\boldsymbol{u}))\\ &\quad + E[h^*(\boldsymbol{u}^l_{s^*(\boldsymbol{u})+1}, \boldsymbol{V}^Z) - h(\boldsymbol{b}^l_{s^*(\boldsymbol{u})+1}, \boldsymbol{V}^Z)]\\ &= E\left[\sum_{i=0}^{\tau} g(\boldsymbol{u}^{(i)}, s^*(\boldsymbol{u}^{(i)})) - g(\boldsymbol{b}^{(i)}, s^*(\boldsymbol{u}^{(i)}))\right].\end{aligned} \tag{9}$$

Since $g(\boldsymbol{u}^{(i)}, s^*(\boldsymbol{u}^{(i)})) - g(\boldsymbol{b}^{(i)}, s^*(\boldsymbol{u}^{(i)})) \ge 0$ for all i, $h^*(\boldsymbol{u}) - h^*(\boldsymbol{b}) \ge 0$.

(iii) Once more we use $\boldsymbol{u}$ instead of $\boldsymbol{b}'$ for notational convenience. The upper bound is easy to show as the policy is restricted to $s > l(\boldsymbol{b})$. To prove $h^*(\boldsymbol{u}) \le h^*(\boldsymbol{b}, \boldsymbol{u})$, we follow a similar proof to that in (ii). The only difference is; the process starting at $\boldsymbol{q} := (\boldsymbol{b}, \boldsymbol{u})$ will be iterated with its respective optimal policy while the one starting at $\boldsymbol{u}$ will be iterated as follows: If $s^*(\boldsymbol{q}^{(i)}) - l(\boldsymbol{b}) < s^*(\boldsymbol{u}^{(i)})$ then take the first one, else take the optimal one. The two processes will again coincide in a finite time τ. Now write down the difference as in proof of (ii):

$$h^*(\boldsymbol{q}) - h^*(\boldsymbol{u}) \ge E\left[\sum_{i=0}^{\tau-1} g(\boldsymbol{q}^{(i)}, s^*(\boldsymbol{q}^{(i)})) - g(\boldsymbol{u}^{(i)}, 1)\right]. \tag{10}$$

From (i), it follows that $\boldsymbol{u}^{(i)}$ is a suffix of $\boldsymbol{q}^{(i)}$ for all i, i.e., $\boldsymbol{q}^{(i)} = (\boldsymbol{u}^{(i)}, \boldsymbol{b}^{(i)})$ for some $\boldsymbol{u}^{(i)}$ with probability 1 and $g(\boldsymbol{q}^{(i)}, s^*(\boldsymbol{q}^{(i)})) \ge g(\boldsymbol{u}^{(i)}, 1)$. Hence, $h^*(\boldsymbol{q}) - h^*(\boldsymbol{u}) \ge 0$.

(iv) Suppose $s^*(\boldsymbol{b}) \neq l(\boldsymbol{b})$ with $v_{s^*(\boldsymbol{b})} = v_{\min}$ and let $\tilde{s}(\boldsymbol{b}) > s^*(\boldsymbol{b})$ be the first index with $v_{\tilde{s}(\boldsymbol{b})} > v_{\min}$. If there is no such index, take $\tilde{s}(\boldsymbol{b}) = l(\boldsymbol{b})$. Again, iterate two processes starting from $\boldsymbol{b}$ with the first one choosing $s^*(\boldsymbol{b})$ and the second one choosing $\tilde{s}(\boldsymbol{b})$ at the first step. Then iterate the two processes as in the proof of (iii) until they coincide at a finite random time τ.

$$\begin{aligned}\boldsymbol{b} = \boldsymbol{b}^{(0)} &\overset{s^*(\boldsymbol{b})}{\to} \boldsymbol{b}^{(1)} \to \boldsymbol{b}^{(2)} \to \ldots \to \boldsymbol{b}^{(\tau)}\\ \boldsymbol{b} = \boldsymbol{u}^{(0)} &\overset{\tilde{s}(\boldsymbol{b})}{\to} \boldsymbol{u}^{(1)} \to \boldsymbol{u}^{(2)} \to \ldots \to \boldsymbol{u}^{(\tau)} = \boldsymbol{b}^{(\tau)}\end{aligned} \tag{11}$$

By the optimality condition we need

$$\begin{aligned}&g(\boldsymbol{b}, \tilde{s}(\boldsymbol{b})) - g(\boldsymbol{b}, s^*(\boldsymbol{b}))\\ &+ E[h(\boldsymbol{b}^{l(\boldsymbol{b})}_{\tilde{s}(\boldsymbol{b})+1}, \boldsymbol{V}^Z) - h(\boldsymbol{b}^{l(\boldsymbol{b})}_{s^*(\boldsymbol{b})+1}, \boldsymbol{V}^Z)] \ge 0\end{aligned} \tag{12}$$

and hence

$$\begin{aligned}&g(\boldsymbol{b}, \tilde{s}(\boldsymbol{b})) - g(\boldsymbol{b}, s^*(\boldsymbol{b}))\\ &+ E\left[\sum_{i=1}^{\tau} g(\boldsymbol{u}^{(i)}, 1) - g(\boldsymbol{b}^{(i)}, s^*(\boldsymbol{b}^{(i)}))\right] \ge 0.\end{aligned} \tag{13}$$

The expression above can be written as

$$\eta(s^*(\boldsymbol{b})-\tilde{s}(\boldsymbol{b}))+\frac{1}{\mu}(v_{\min}-v_{\tilde{s}(\boldsymbol{b})}) + E\Bigg[\frac{1}{\mu}\sum_{i=1}^{\tau}(\boldsymbol{b}^{(i)}_{s^*(\boldsymbol{b}^{(i)})}-\boldsymbol{u}_1^{(i)}) + \eta\sum_{i=1}^{\tau}(l(\boldsymbol{u}^{(i)})-1-l(\boldsymbol{b}^{(i)})+s^*(\boldsymbol{b}^{(i)}))\Bigg]. \quad (14)$$

Property 1, (i) implies $l(\boldsymbol{u}^{(i)})-1-(l(\boldsymbol{b}^{(i)})-s^*(\boldsymbol{b}^{(i)}))\leq 0$ for all $i\leq\tau$ and we also have $\sum_{i=1}^{\tau}(\boldsymbol{b}^{(i)}_{s^*(\boldsymbol{b}^{(i)})}-\boldsymbol{u}_1^{(i)})\leq 0$. Since $v_{\min}-v_{\tilde{s}(\boldsymbol{b})}<0$ and $s^*(\boldsymbol{b})-\tilde{s}(\boldsymbol{b})<0$, the above expectation is negative. Hence we obtained a contradiction with (12) and the statement is proved.

C. Proof of Theorem 2

For simplicity assume $v_{\min}=1$. Let $v:=v_{\max}$. Take any state space $\mathcal{V}^{\leq K'}$ with $K'>K=K(\eta)$ and start the policy iteration algorithm with an initial policy $\boldsymbol{s}^{(0)}$ such that $l(\boldsymbol{b})-s^{(0)}(\boldsymbol{b})<K$ and $s^{(0)}(\boldsymbol{b})$ never takes 1 if $s^{(0)}(\boldsymbol{b})\neq l(\boldsymbol{b})$ for all $\boldsymbol{b}\in\mathcal{V}^{\leq K'}$. We will show that the policy iteration cannot violate the condition $l(\boldsymbol{b})-s^{(i)}(\boldsymbol{b})<K$ for any i. Verbally, this means the updated policy will not send any data generated more than K slots ago. This concludes that a buffer size of K is sufficient for the optimal policy. Note that it is enough to show that this condition holds for $i=1$.

Now start the algorithm with such $\boldsymbol{s}^{(0)}$ described above, which gives a relative value vector $\boldsymbol{h}^{(0)}=\boldsymbol{h}$. Assume $\boldsymbol{h}$ satisfies Property 1 (i) and (ii) as there is an $\boldsymbol{s}^{(0)}$ satisfying these and the conditions above, e.g., $\boldsymbol{s}^{(0)}=l(\boldsymbol{b})$. Also note that Property 1 (i) and (ii) hold after the first policy iteration. Denote $\boldsymbol{s}^{(1)}=\tilde{\boldsymbol{s}}$. Take any state $\boldsymbol{b}\neq 1^{l(\boldsymbol{b})}$ with respective policies $s^{(0)}(\boldsymbol{b})$, $\tilde{s}(\boldsymbol{b})$ and $l(\boldsymbol{b})>K$. We omit the argument $\boldsymbol{b}$ in the above expressions for simplicity. Define $s:=\min\{j>l-K: b_j=v\}$, i.e., index of the first occurrence of v in the most recent K time slots.

Observe that if $l-\tilde{s}\geq K$, the following must hold (by step 3 of Algorithm 1):

$$\eta(s-\tilde{s})+E[h(\boldsymbol{b}^l_{\tilde{s}+1},\boldsymbol{V}^Z)-h(\boldsymbol{b}^l_{s+1},\boldsymbol{V}^Z)]\leq\frac{1}{\mu}\sum_{j=\tilde{s}}^{s-1}b_j \quad (15)$$

Since $l(\boldsymbol{b})-s^{(0)}(\boldsymbol{b})<K$ for all $\boldsymbol{b}\in\mathcal{V}^{\leq K'}$, the current policy cannot choose any content generated in further past from K time units. Hence, $h(\boldsymbol{b}^l_{\tilde{s}+1},\boldsymbol{V}^Z)=\frac{1}{\mu}\sum_{j=\tilde{s}+1}^{l-K}b_j+h(\boldsymbol{b}^l_{l-K+1},\boldsymbol{V}^Z)$. Moreover, from the definition of s, we know that $b_j=1$ for $l-K+1\leq j\leq s-1$. The LHS of the above condition is thus equivalent to

$$\begin{aligned}&\eta(s-\tilde{s})+E[h(\boldsymbol{b}^l_{l-K+1},\boldsymbol{V}^Z)-h(\boldsymbol{b}^l_{s+1},\boldsymbol{V}^Z)]\\&=\eta(s-\tilde{s})+E[h(1^{s+K-l-1}\boldsymbol{b}^l_s,\boldsymbol{V}^Z)-h(\boldsymbol{b}^l_{s+1},\boldsymbol{V}^Z)]\\&=\eta(s-\tilde{s})+\frac{1}{\mu}(s+K-l-1)\\&\quad+E[h(\boldsymbol{b}^l_s,\boldsymbol{V}^Z)-h(\boldsymbol{b}^l_{s+1},\boldsymbol{V}^Z)],\end{aligned}$$

and the RHS is equivalent to

$$\frac{1}{\mu}(b_{\tilde{s}}+s+K-l-1).$$

Hence, (15) is equivalent to

$$\eta(s-\tilde{s})+E[h(\boldsymbol{b}^l_s,\boldsymbol{V}^Z)-h(\boldsymbol{b}^l_{s+1},\boldsymbol{V}^Z)]\leq\frac{1}{\mu}b_{\tilde{s}}$$

and using $\tilde{s}\leq l-K$ and $b_{\tilde{s}}\leq v$, we obtain a weaker condition

$$\eta(s+K-l)+E[h(\boldsymbol{b}^l_s,\boldsymbol{V}^Z)-h(\boldsymbol{b}^l_{s+1},\boldsymbol{V}^Z)]\leq\frac{1}{\mu}v \quad (16)$$

which is the same as checking if the following expectation is negative or not.

$$E\left[\eta(s+K-l)+h(\boldsymbol{b}^l_s,\boldsymbol{V}^Z)-h(\boldsymbol{b}^l_{s+1},\boldsymbol{V}^Z)-\frac{1}{\mu}v\right]$$

Define $R:=h(\boldsymbol{b}^l_s,\boldsymbol{V}^Z)-h(\boldsymbol{b}^l_{s+1},\boldsymbol{V}^Z)=h(v,\boldsymbol{b}^l_{s+1},\boldsymbol{V}^Z)-h(\boldsymbol{b}^l_{s+1},\boldsymbol{V}^Z)$ and $R':=\eta(s+K-l)+R-\frac{1}{\mu}v$. We want to lower bound $E[R']$. Consider the conditional expectation $E[R|\boldsymbol{V}^Z=\boldsymbol{v}^z]=E[h(v,\boldsymbol{b}^l_{s+1},\boldsymbol{v}^z)-h(\boldsymbol{b}^l_{s+1},\boldsymbol{v}^z)|\boldsymbol{V}^Z=\boldsymbol{v}^z]$. Similar to the previous proofs, consider two coupled processes $\{(\boldsymbol{u}^{(i)},\boldsymbol{q}^{(i)})\}$ starting from $(v,\boldsymbol{b}^l_{s+1},\boldsymbol{v}^z)$ and $(\boldsymbol{b}^l_{s+1},\boldsymbol{v}^z)$ respectively and both iterated according to $\boldsymbol{s}^{(0)}$.

Define τ as the stopping time defined with the following stopping condition: (A1) the two processes coincide, i.e, $l(\boldsymbol{u}^{(\tau)})-s^{(0)}(\boldsymbol{u}^{(\tau)})=l(\boldsymbol{q}^{(\tau)})-s^{(0)}(\boldsymbol{q}^{(\tau)})$, or (A2) the second process $\boldsymbol{q}$ chooses the data at tail, i.e., $s^{(0)}(\boldsymbol{q}^{(\tau)})=l(\boldsymbol{q}^{(\tau)})$, and they do not coincide. Denote the events in condition (A1) and (A2) as $\mathcal{A}_1$ and $\mathcal{A}_2$. We note that $\mathcal{A}_1\cap\mathcal{A}_2=\emptyset$. Furthermore, τ is finite with probability 1. This can be seen as $\boldsymbol{q}$ will consist only of 1s eventually and sends the one at the tail. We observe that for $0\leq i\leq\tau$, $\boldsymbol{q}^{(i)}$ is a suffix of $\boldsymbol{u}^{(i)}$ from Property 1, (i).

Now observe that $\mathcal{A}_1$ only occurs if the prefix of $\boldsymbol{u}^{(\tau)}$ before $\boldsymbol{q}^{(\tau)}$ contains a v. If it does not contain any v, it means that $\boldsymbol{q}^{(\tau-1)}$ consists only of 1s and thus sends the data at the tail; which implies that the iterations must have been stopped before. If $\mathcal{A}_1$ occurs, this means that the first process must have missed a v, as it is in the prefix and not taken, and hence $R\geq\frac{1}{\mu}v$.

If $\mathcal{A}_2$ occurs, then (conditioned on $\boldsymbol{V}^Z=\boldsymbol{v}^z$)

$$\begin{aligned}R=&\sum_{i=0}^{\tau}g(\boldsymbol{u}^{(i)},s^{(0)}(\boldsymbol{u}^{(i)}))-g(\boldsymbol{q}^{(i)},s^{(0)}(\boldsymbol{q}^{(i)}))\\&+E[h(\boldsymbol{u}^{(\tau+1)})-h(\boldsymbol{q}^{(\tau+1)})].\end{aligned}$$

As the two processes have not coincided yet, $\boldsymbol{q}^{(\tau+1)}$ is still a suffix of $\boldsymbol{u}^{(\tau+1)}$ and therefore $E[h(\boldsymbol{u}^{(\tau+1)})-h(\boldsymbol{q}^{(\tau+1)})]\geq E[h(1,\boldsymbol{q}^{(\tau+1)})-h(\boldsymbol{q}^{(\tau+1)})]\geq\frac{1}{\mu}$. This is seen by combining Property (ii) and (iv). Further note that

$$\sum_{i=0}^{\tau}g(\boldsymbol{u}^{(i)},s^{(0)}(\boldsymbol{u}^{(i)}))-g(\boldsymbol{q}^{(i)},s^{(0)}(\boldsymbol{q}^{(i)}))\geq\eta(l-s)$$

as $\boldsymbol{q}^{(i)}$ remains a suffix of $\boldsymbol{u}^{(i)}$ until τ and age penalty in the above expression is minimized when $\boldsymbol{u}$, at $(i+1)^{\text{th}}$ iteration, chooses the same data that $\boldsymbol{q}$ chooses at i^{th} iteration. Hence, $R\geq\frac{1}{\mu}+\eta(l-s)$.

Finally, we write

$$\begin{aligned}
E[R'|\boldsymbol{V}^Z = \boldsymbol{v}^z] &= E[R'; \mathcal{A}_1|\boldsymbol{V}^Z = \boldsymbol{v}^z] + E[R'; \mathcal{A}_2|\boldsymbol{V}^Z = \boldsymbol{v}^z] \\
&\geq \Pr(\mathcal{A}_1)\left(\frac{1}{\mu} - \frac{1}{\mu}\right) + \Pr(\mathcal{A}_2)\left(\eta K - \frac{1}{\mu}(v-1)\right) \\
&\geq \Pr(\mathcal{A}_2)\left(\eta K - \frac{1}{\mu}(v-1)\right) \\
&\geq 0
\end{aligned}$$

as the choice of K is such that $\eta K \geq \frac{1}{\mu}(v-1)$. Hence, $E[R'] \geq 0$. The weaker condition (16) does not hold and thus (15) can never hold. This concludes that $\tilde{s} = s^{(1)}(\boldsymbol{b})$ can never be smaller than $l(\boldsymbol{b}) - K$ for $\boldsymbol{b} \neq 1^{l(\boldsymbol{b})}$.

The case for $\boldsymbol{b} = 1^L$, $K' \geq L > K$ is straightforward. For $s^{(1)}(\boldsymbol{b}) < K' - L$, one needs $\eta(K+1) \leq \eta(L - s^{(1)}(\boldsymbol{b})) \leq \frac{v-1}{\mu}$, which is not true.