Achievable Optimal Age of Information with Segmentation and Predictive Scheduling Policy

Jin Zhang, Peng Zou, Suresh Subramaniam
ECE Department, George Washington University, Washington DC, 20052, USA
zhangjin, pzou94, suresh@gwu.edu

***Abstract*—As the metric of Age of Information (AoI) has gained its popularity in recent years, a collection of scheduling policies has been proposed in order to optimize AoI. It is natural to raise the question of how much these scheduling policies could be improved and how many predictive packets are necessary to achieve optimum. The performance of scheduling policies can be quantitatively evaluated by comparing them with the optimal AoI. Given a packet sequence, there must be at least one packet combination to achieve optimal AoI, which can be obtained by exhausting all possible decisions of preserving/rejecting packets. However, it is not effortless to attain prior knowledge of each update's arrival and service time. Moreover, this simple scheduling policy of traversing every combination of 2^n consumes unaffordable resources of time and energy and computing power.**

Through mathematical analysis, we found that, given a sufficiently long packet sequence, it could be segmented into local epochs with invariant global optimal policy, and local optimization of each epoch incrementally aggregates the global optimal AoI of the entire sequence. From this new perspective, this also explains the counter-intuitive phenomenon [1] [2] [3] of idle waiting time instead of transmitting updates immediately. By taking advantage of the segmentation proposed in this paper, we only utilize part of inter-arrival times and service times of a finite number of future updates to acquire the global optimal AoI for a full sequence. Numerical results also show that under exponential distribution, the segmentation algorithm with partial prediction needs only to predict 2-10 subsequent updates in future and accurately achieve the optimal AoI. After comparing with the AoIs obtained from other scheduling policies, optimization performance of predictive scheduling policy prevails over others.

I. Introduction

To explore optimal policies that can guarantee timeliness of wireless network, researchers recently proposed a novel metric AoI (age of information) [1] and its derivations peak age [2] [5], overage probability [4] [9], Max-AoI [8] [14] and value of information [15]. Given the probability distributions of service time and arrival time of packets, the threshold policies [7] [13] [12] achieved their average optimal AoIs in statistics. In addition, some researchers have also proposed deterministic policies with fixed-probability [6] [7] and obtained an optimal scheduling probability to determine whether to abandon or preserve being-served packets. These two categories of policies for improving wireless networking performance can make determinations with distribution knowledge to ensure that AoI reaches a certain degree of optimization. Nevertheless, these policies are featured with average-based AoI optimization in statistics rather than accurate optimum.

A. Motivation and Problem Statement

At present, there has been no strict theoretical basis regarding evaluating which of these policies have better performance since the disadvantage of the optimization policies in statistics is concerned with introducing predictive policy. For instance, the threshold-based optimization policy derives fixed-threshold from existing information of packet sequence [13] and explores the optimal peak AoI [2]. Once the service time exceeds the fixed threshold, due to particular packets with lengthened service times, the scheduling policy takes actions (such as an deterministic policy) to preempt those timeout packets and ensure that the information age does not evolve excessively and even diverge as time increases. It is supposed to be continuing and completing this transmission session with a bit of timeout for benefit of long-term AoI deduction, when a fixed threshold is triggered with a small amount of timeout. However, traditional preemptive scheduling policies [7] [13] would abandon this timeout packet with threshold constraint violation that should have been delivered to server within a short period of time. As a result, the corresponding AoI value that could have been smaller becomes greater instead. Unfortunately, it cannot achieve the perfect effect of optimizing transmission in this case. In addition, the predefined configuration of fixed threshold requires prior knowledge of the entire packets' distribution, which is not precisely obtained in advance. Moreover, the threshold based on overall distribution can lead to a slower response speed for immediate optimization with respect to local-based policy. As in alternative words, the threshold-based and fixed-probability scheduling policies have flaws. Although these threshold-based approaches on wireless network and edge computing may obtain average optimal in statistics and near-optimal in theoretic for a given sequence of packets, they cannot achieve the optimal AoI.

There are difficulties in finding optimal AoI for a long sequence in mobile communication with resource constraints. The total number of packets in transmission tasks could be significant (such as millions and more). Furthermore, the exponential growth of possible combinations of 2^n makes a simple exhaustive scheduling policy impossible to implement in a wireless circumstance.

B. Solution and Contribution

A theorem in this article reveals that one or more idle states (called separator) exist between the optimal packet distributions in the FIFO queue system with infinite buffer. It divides a single long sequence into multiple short sequences (defined as epochs) and superimposes them to reconstruct a full optimal AoI process. The intuition originates from the observation that the total transmission time of overall packets tends to be much longer than single inter-arrival and service time. Some works have made progress on inter-arrival time prediction [10] and also contribute to our long sequence segmentation approach. Specifically, our contributions are outlined as follows.

- We propose a partial predictive approach of segmenting an arbitrary packet sequence following an exponential distribution. By applying queuing system model, we formulate general mathematical expressions for calculating AoI for any scheduling policy in a system with infinite buffer. Theorems provide that our algorithm divides long sequence into short epochs, by proving the properties of long sequence segmentation and separator periodicity.
- Only finite number of packets are necessary to obtain optimum with respect to global optimal AoI. Predictive scheduling policy with local packets achieves global optimal AoI with partial prediction. More future packets are predicted and higher optimization accuracy it reaches.
- This article analyzes fault tolerance for packet prediction error and evaluates the impact of estimation errors on achieving optimal AoI. Experiments shows that limited prediction error does not affect global optimum as well as error range results in parts of incorrect decision making and deviation from the optimal AoI to some extent.
- Simulation demonstrates quantitative comparison results with partial or full prediction-based and fixed probability-based average age. The probability of separator presence reaches up to 99% and even higher within ten packets.

The remainder of this article is organized as follows. The system model is introduced in Section 2. In Section 3, optimization problem is formulated. In Section 4, by theoretical analysis, we present the segmentation mechanism. In section 5 and 6, the predictive scheduling policy with fault tolerance and numerical results are provided respectively.

II. System Model

A point-to-point communication system is considered with a single sender-receiver pair transmitting status updates from a source to a destination. A predictor estimates inter-arrival time [1] $I_i = t_i - t_{i-1}$, where t_i is the generation time of the i^{th} packet ($t_0 = 0$), and service time X_i of the future updates [10] to determine whether to discard or preserve the update under being serviced, as shown in Fig. 1. The model has an infinite buffer with first-come first-served (FCFS) that allows the system to store all updates generated by a source. It also assumes that future updates' inter-arrival time and service time can be predicted either fully or partially.

III. Optimization Problem Formulation

In the model with full packet prediction, a simple scheduling policy exhausts every possible combination and achieve the optimal AoI process. However, due to huge search space $\pi \in$

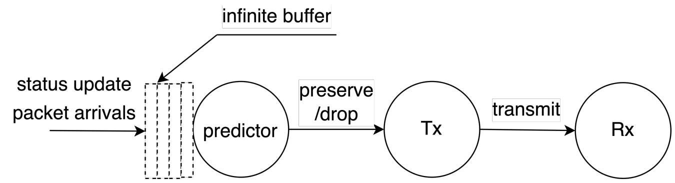

Fig. 1. System model with status update packets arriving at a single server with queues (infinite buffer), through predictor and scheduler determining to either preserve or drop updates and minimizing the age of information.

$\prod$ and resource consumption of exhaustive scheduling policy, we propose segmentation approach.

A. Symbol declarations and variable definitions

The initial age $A_0^{(i)}$ of an AoI process represents the initial AoI value at the very beginning instant of a process, with which the next new process begins. The final age $A_{fin}^{(i)}$ of an AoI process denotes the remains of an AoI process. The main notation table is given in the online article at https://www.overleaf.com/read/yvhwcxrchfjk. The age of information for the i^{th} packet evolves as

$$A(t) = t - t_i, \qquad i \in \{1, 2, \dots, N\}$$

An idle waiting time among long sequence is defined as an explicit or implicit separator and emerges periodically. The original distribution generates an idle state called *explicit separator*. There is no idle waiting time between originally distributed updates, but an idle state exists after optimization and it is regarded as *implicit separator*, as shown in Fig.2(a).

B. Problem formulation

The delivery of each packet dynamically determines an AoI evolution process. The packet delivery timing equation under an arbitrary policy π for each update is derived as

$$d_i^{\pi} = t_i' - t_i = \tau_i^{\pi} + X_i = \sum_{q=i-k}^{i-1} (\tau_q^*)^{\pi} + X_i \tag{1}$$

where the τ_i^{π} denotes delay time of the i^{th} packet and $(\tau_q^*)^{\pi} = X_q - I_q$ represents multiple overlapping *equivalent delay times* of k updates and contributes to the current i^{th} update's delay time, under any scheduling policy π, since the nearest idle waiting time at the $(i-k)^{th}$ update, where $i \geq 1$, $k \geq 0$ and $i - k \geq 1$.

As shown in the Fig.2(b). the general evolution equation [1] [11] and the corresponding local cumulative AoI Φ_n for each packet after inter-arrival time I_n is calculated as

$$\Phi_n = I_n d_n + \frac{1}{2} I_n^2, \qquad n \in \{1, 2, \dots, N\} \tag{2}$$

Every element in the ordinal vector $\overrightarrow{\mathbf{l}^{(1)}} \in \{l_1^{(1)}, l_2^{(1)}, ..., l_m^{(1)}\}$ sequentially denotes each index number $l_x^{(1)}$ of successful transmitted updates. The number of delivered updates m and the total number of updates k generated by source necessarily satisfies the relationship $m \leq k$ in an epoch. By rewriting a decimal variable in binary form, for any scheduling policy

$\pi \leftarrow \{b_1 b_2 ... b_k\}$, we define an auxiliary intermediate variable b_x as each decision for every update when the '1' stands for transmission permission and '0' for packet rejection. The value b_x for each decision is defined as

$$b_x = \begin{cases} 0 & x \notin \overrightarrow{\mathbf{l}^{(1)}} \\ 1 & x \in \overrightarrow{\mathbf{l}^{(1)}} \end{cases} \qquad x \in \{1, 2, ..., k\}$$

Employing any scheduling policy π, we have the age of information $\Phi_n^{\pi}(k)$ for k^{th} update indexed from $(i_n+k-1)^{th}$ to $(i_n+k)^{th}$ shown in the Fig.2(b). An age sample path shows updates in an epoch $(i_n, i_n+1, ..., i_n+k)$ arrive at source and are delivered at server. But it discards the update at (i_n+1).

By substituting multiple inter-arrivals $I_n = \sum_{s=l_x^{(1)}}^{l_{x+1}^{(1)}} I_{i_n+s}$ and latest delivery time $d_n = d^{\pi}_{i_n+l_{x+1}^{(1)}}$ into the equation (2), when discarding multiple consecutive updates between $[l_{x+1}^{(1)} - l_x^{(1)}]$, we have generic equation for two preserving actions at $l_x^{(1)}$ and $l_{x+1}^{(1)}$ of any policy π, illustrated in the Fig.2(b).

$$\Phi_n^{\pi}(l_x^{(1)}) = \left(\sum_{s=l_x^{(1)}}^{l_{x+1}^{(1)}} I_{i_n+s} \right) d^{\pi}_{i_n+l_{x+1}^{(1)}} + \left(\sum_{s=l_x^{(1)}}^{l_{x+1}^{(1)}} I_{i_n+s} \right)^2 \Big/ 2 \tag{3}$$

By adding together local cumulative AoI $\Phi_n^{\pi}(l_x^{(1)})$ for every update in an epoch, which is defined in equation (3), then it gives a general AoI equation of an epoch under policy π.

$$Q_n^{\pi}(i_n, k_n) = Q_n^{\pi}(i_n, k_n, \vec{\mathbf{b}_n}) = \sum_{x=1}^{m} \Phi_n^{\pi}(l_x^{(1)}) =$$

$$\sum_{x=1}^{m} \left[\left(\sum_{s=l_x^{(1)}}^{l_{x+1}^{(1)}} I_{i_n+s} \right) d^{\pi}_{i_n+l_{x+1}^{(1)}} + \left(\sum_{s=l_x^{(1)}}^{l_{x+1}^{(1)}} I_{i_n+s} \right)^2 \Big/ 2 \right] \tag{4}$$

where the policy decision vector $\pi \leftarrow \vec{\mathbf{b}_n} = b_{n1} b_{n2} b_{n3} ... b_{nk_n}$, $b_{nj} \in \{0, 1\}, \forall j \in \{1, 2, ..., k_n\}$ for the n^{th} epoch.

The objective function $\mathbb{E}[J^{\pi^*}]$ of minimizing average AoI in search space $\pi \in \prod$ when the total transmission time $T = \sum_{h=1}^{\infty} I_h$ approximates toward infinity is formulated

$$\mathbb{E}[J^{\pi^*}] = \mathbb{E}\left[\sum_{n=1}^{\infty} J_n^{\pi_n^*} \right] = \underset{\pi \in \prod}{\text{minimize}} \left\{ \frac{\sum_{n=1}^{\infty} Q_n^{\pi_n}(i_n, k_n, \vec{\mathbf{b}_n})}{\sum_{h=1}^{\infty} I_h} \right\} \tag{5}$$

We will apply sequence segmentation to the objective function (5) and find the optimal policy π^*, decision vector $\vec{\mathbf{b}}^*$.

IV. AoI Optimization with Segmentation

The equivalence of global optimal to a summation of local optimums can be mathematically expressed as follows.

$$\sum_{n=1}^{\infty} \left\{ \min_{\pi_n \in \prod_n^*} \left[Q_n^{\pi_n}\left(i_n, k_n\right) \right] \right\} = \min_{\pi \in \prod_{n=1}^*} \left[\lim_{k_1 \to \infty} Q_{n=1}^{\pi}\left(1, k_1\right) \right]$$

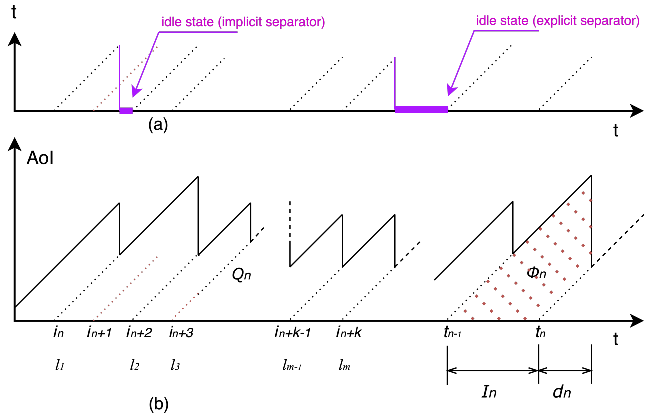

Fig. 2. (a) Original distribution drives the age at server to evolve as time increases. (b) updates $(i_n, i_n+1, ..., i_n+k)$ in the n^{th} epoch from a source arrive at transmitter at times $t_{i_n}, t_{i_n+1}, ..., t_{i_n+k}$ and are received by server at times $t'_{i_n}, t'_{i_n+1}, ..., t'_{i_n+k}$.

Theorem 1 *With respect to optimal AoI, the local optimal scheduling policies for a segmented epoch are exact the same ones as the global optimal for the entire long sequence, when the update sequence formulation in the epoch simultaneously satisfies the following three conditions:*

1) *minimum AoI value out of all possible policies,*
2) *minimum final age out of all possible policies,*
3) *idle state recurrently appears between optimal epochs.*

To begin with, we present the other two auxiliary theorems (Theorem 2 and 3) to prove segmentation Theorem 1, by showing how to reconstruct global optimal with sub-policies.

Theorem 2 *For two different initial ages, the sequence with smaller initial age $A_0^{(1)}$ has less optimal AoI value $Q^{\pi_1^*}(A_0^{(1)})$ than with greater $A_0^{(2)}$ and optimal AoI value $Q^{\pi_2^*}(A_0^{(2)})$ under the same inter-arrival and service times. The function of minimum AoI $y = Q^{\pi^*}(x)$ is monotonic decreasing.*

$$Q^{\pi_1^*}(A_0^{(1)}) \leq Q^{\pi_2^*}(A_0^{(1)}) \leq Q^{\pi_2^*}(A_0^{(2)}) \tag{6}$$

where π_1^*, π_2^* are optimal policies for $A_0^{(1)}$, $A_0^{(2)}$ respectively.

Here, it is the proof of Theorem 2. An AoI value for any sequence is determined by four factors: initial age, arrival time, service time and scheduling policy. The first part of inequity (6) must hold $Q^{\pi_1^*}(A_0^{(1)}) \leq Q^{\pi_2^*}(A_0^{(1)})$ because both sides have the same packet sequence and initial age but different policies π_1^* and π_2^*. According to the definition of optimal age of information, we know that $Q^{\pi_1^*}(A_0^{(1)})$ under policy π_1^* must be the minimum among all possible policies for a given packet sequence and a specific initial age. For the second part of the inequity, both sides adopt same policy π_2^* and packet sequence but different initial age. It is evident that the sequence with less initial age has less optimal AoI value than that with greater initial age. Finally, the inequity (6) holds.

We will prove that separator inevitably and periodically appears in an optimal policy. This property also explains the counter-intuitive phenomenon [1] [2] [3] of idle waiting time existing between updates. Let $q_i^\pi(t)$ be the number of packets in queue i at the beginning of time t when policy π is employed. Then, it guarantees that queue i is stable if $\lim_{T\to\infty} \mathbb{E}[q_i^\pi(T)] \le \infty$ [2]. The equivalent inequality of system stability τ_n is expressed as $\lim_{n\to\infty}\left[\sum_{i=n-c}^{n}\left(\tau_i\right)^\pi\right] = M_n^\pi \le \infty$ where c denotes the number of delayed packets since the nearest idle state, the finite value M_n^π represents the maximum limit of delay time for an arbitrary packet at queuing.

Theorem 3 *A sub-sequence, starting from an arbitrary update of an infinite update sequence and containing all the rest of consecutive updates, must have explicit or implicit separators between optimized updates with respect to optimal global AoI.*

For more details, the proof of the above theorem 3 is given online at https://www.overleaf.com/read/yvhwcxrchfjk.

By the theorem 3, separator existence is guaranteed to divide a long sequence into epochs without affecting optimal AoI. As given in the theorem 2, AoI evolving from minimum initial age $A_0^{(1)}$ and it always satisfies $Q_n^{\pi_1^*}(A_0^{(1)}) \le Q_n^{\pi_1^{[m]}}(A_0^{(m)})$. Additionally, the final age $A_F^{(1)}$ of the local optimal policy is also less than other final ages $A_F^{(m)}$ under any other local policies and $Q_{n+1}^{\pi_2^*}(A_F^{(1)}) \le Q_{n+1}^{\pi_2^{[m]}}(A_F^{(m)})$. Correspondingly, $Q_{n+k}^{\pi_k^*}(A_F^{(1)}) \le Q_{n+k}^{\pi_k^{[m]}}(A_F^{(m)})$ holds. The theorem 1 is proved.

V. Limited Prediction and Fault Tolerance

This chapter analyzes fault tolerance of prediction error in order to satisfy the timeliness requirement and average deviation ΔA_{ave}^* between estimated and optimal AoIs is expressed as $|\Delta A_{ave}^*| \le a_{max}^*$. With segmentation, we propose *N-prediction* algorithm for limited number of predictable packets.

Packets are classified as four types, the number p of earlier packets $\{I_{n-p}^{(n-p,n-1)}, X_{(n-p)}^{(n-p,n-1)}\}$ in the past, being transmitted packet $\{I_n, X_n\}$, predictable future packets $\{I_n^{(n,k)}, X_n^{(n,k)}\}$ and unpredictable future packets $\{I_{(n+k)}^{(n+k,l)}, X_{n+k}^{(n+k,l)}\}$. The estimation errors for inter-arrival $\tilde{I}_{n+k}^{(n+k,l)}$ and system times $\tilde{X}_{n+k}^{(n+k,l)}$ are respectively defined as $\delta_I^{(k,l)}$ and $\varepsilon_X^{(k,l)}$ following uniform distribution of $[-\delta_{max}, +\delta_{max}]$ and $[-\varepsilon_{max}, +\varepsilon_{max}]$.

Each estimated decision $\tilde{b}_n$ of preserving or discarding current packet, the distance ΔA_n^* between optimal AoI and limited predictive policy as well as the long-term average deviation ΔA_{ave}^* are expressed as

$$\tilde{b}_n = \underset{(k,l)}{\operatorname{argmin}}\{\mathrm{Q}_\mathrm{n}^\pi(I_n^{(n,k)}, X_n^{(n,k)}, \tilde{I}_{n+k}^{(n+k,l)}, \tilde{X}_{n+k}^{(n+k,l)})\}$$

$$\Delta Q_n^* = \left[Q_n^{\tilde{\pi}_n^*}(i_n, k_n, b_n) - Q_n^{\pi_n^*}(i_n, k_n, b_n)\right]$$

$$\Delta A_{ave}^* = \lim_{N\to\infty}\left\{\sum_{n=1}^{N}\Delta Q_n^* \Big/ \sum_{n=1}^{N} I_n\right\}$$

The objective function in *N-prediction* algorithm is to minimize long-term total average deviation.

$$\limsup_{n\to\infty}\{\Delta A_{ave}^*\} = \min_{\forall s\in S_j}\lim_{N\to\infty}\left\{\sum_{n=1}^{N}\Delta Q_n^* \Big/ \sum_{n=1}^{N} I_n\right\} \quad (7)$$

The theorem 4 presents that deviation can also be divided into epochs without jeopardizing total deviation in a long-term.

Theorem 4 *After segmentation with partial estimated decisions, the average minimum deviation for all epochs in long term converges to zero and it is mathematically expressed as*

$$\lim_{N\to\infty}\left\{\Delta A_{ave}^*\middle|\left(\tilde{\pi}_n^{(k,l)}, \pi_n^{(k,l)}\right)\right\} = 0 \quad (8)$$

where the N-prediction policy with estimation error $\tilde{\pi}_n^{(k,l)} = \{b_n, b_{n+1}, ..., \tilde{b}_{n+k}, \tilde{b}_{n+k+1}, ..., \tilde{b}_{n+k+l-1}, b_{n+k+l}, ...\}$ *and the optimal policy with accurate prediction* $\pi_n^{(k,l)} = \{b_n, b_{n+1}, ..., b_{n+k}, b_{n+k+1}, ..., b_{n+k+l-1}, b_{n+k+l}, ...\}$.

For more details, the archived article online gives its proof at https://www.overleaf.com/read/yvhwcxrchfjk. According to the above theorem 4, the error estimation δ_n^{err} for the n^{th} epoch is mathematically expressed as

$$\delta_n^{err} = Q_n^*(\mathbb{S}_n, b_n) - Q_n^*(\mathbb{S}_n, \tilde{b}_n) \quad (9)$$

The state $\mathbb{S}_n^{(k,l)}$ for an epoch stands for epoch formulation with four types of packets and the notation is expressed as $\mathbb{S}_n^{(k,l)} = \left\{A_0^{(i)}, \left(I_n^{(n,k)}, X_n^{(n,k)}, \tilde{I}_{n+k}^{(n+k,l)}, \tilde{X}_{n+k}^{(n+k,l)}\right)\right\}$.

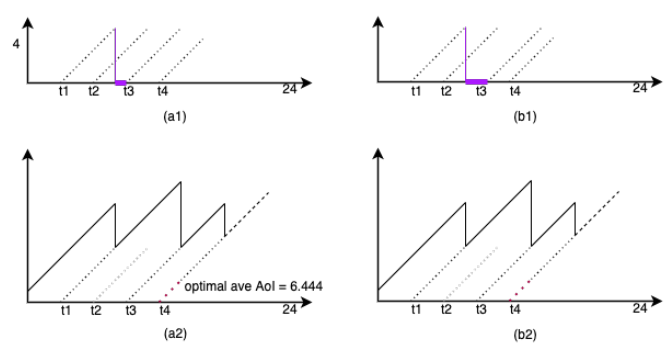

Fig. 3. (a1) original distribution and accurate predictions. (b1) estimated distribution and error predictions. (a2) and (b2) have different distributions and estimated error but both have the same optimal AoI evolution.

By comparing with the optimal AoI and substituting the equation (4) into the above equation (9), we have

$$\left|\left[Q_n^{\tilde{\pi}_n^*}(i_n, k_n, b_n) - Q_n^{\pi_n^*}(i_n, k_n, b_n)\right] \Big/ \sum_{s=i_n}^{i_n+k_n-1} I_s\right| \le a_{max}^* \quad (10)$$

As shown in the Fig.3 , when an estimated future packet does not change optimal policy, it means that the AoI evolution process would achieve optimal value. The deviation between estimated and optimal will be the same and it is expressed as $\Delta Q_n^* = 0$ in this case. By solving inequality (10), we do experiments to statistically obtain the acceptable estimation error range for the assigned requirement of a_{max}^* (e.g. 1.0%).

VI. Predictive Scheduling Policy and Algorithm

This chapter describes how the predictive scheduling policy searches and identifies a separator and then achieves optimum. After starting time of a separator, algorithm traverses packet combinations and finds a natural delivery time of the newly generated packet. From the starting time of separator to the generation time of the first packet, all the packets during this period constitute a new *first attempt epoch*, and the optimal AoI evolution process can be found through exhausting all of packet combinations in this epoch. Then it applies the three criteria of Theorem 1 to determine starting time of a new separator. If a separator is not found at first attempt, the algorithm appends a newly generated packet to the tail of the first attempt epoch to form the *second attempt epoch*. It repeats appending next packet and exhaustively searches as described in previous procedure, until it successfully finds new separator.

In algorithm design, there are two main loops for discovering global optimal, as shown in pseudo-code algorithm 1.

Algorithm 1 Discovering global optimal with segmentation

1: $S_n^{(j=1)} \leftarrow \{X_{n+1}, X_{n+2}, ..., X_{n+k}\}\{I_{n+1}, I_{n+2}, ..., I_{n+k}\}$
2: **while** $n \leq m$ **do** ▷ n,m for epoch index and total number
3:     **for** $i \leftarrow 1, j \leftarrow 1, S_i^{(j)} \leftarrow$ traverse$(S_n^{(j)})$ **do**
4:         **if** A^*_{ave} at idle of $S_i^{(j)}$, $A^{(0)}_{Fin}$ at π^*_n **then**
5:             $S_n^{(j)} \leftarrow S_i^{(j)}$ ▷ epoch found
6:         **else** ▷ append next packet to epoch and continue
7:             $S_i^{(j+1)} \leftarrow S_i^{(j)} + \{X_{n+k+1}\}\{I_{n+k+1}\}$
8:         **end if**
9:     **end for** ▷ i, j for combination and attempt times
10: **end while**
11: return vector $S \leftarrow \{S_1, S_2, ..., S_m\}$

The outer loop consists of epochs, illustrated from lines 2 to 10. The inner loop is designed to explore and identify an epoch from long sequence and achieve the optimal AoI by exhausting all possible combinations from lines 3 to 9. The separator and minimum final age checks are executed at line 4. At last, a return vector delivers discovered optimal epochs.

VII. Numerical Results

This section presents numerical results to explore the optimization performance of the predictive scheduling policy with segmentation and validate our theoretical results. We provide a graphical example of a long sequence transmitting 18 packets from source to destination. The local optimal epochs split by implicit and explicit separators are colored and differentiated by multiple ribbons. We use exhaustive algorithm to find optimal sub-policy for each epoch. As shown in fourth sub-graph at the lowest part of Fig.4, the original distribution forms a sequence. For later comparison, optimal distribution in the third sub-graph is obtained from the fourth.

In the case of the exponential distribution, the simulated graph of explicit and implicit separators statistically appears within a number of packets arriving. A total of 10,000 independent experiments, as shown in Fig. 5 when arrival rate

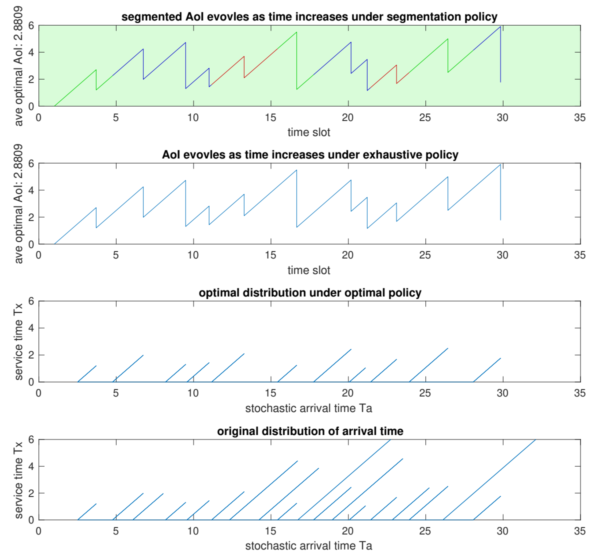

Fig. 4. *The predictive scheduling policy produces optimal AoI in sub-graph 1, while the exhaustive policy produces sub-graph 2 with high intense resource consumption. Moreover, each epoch in various colors is identified in sub-graph 1 and is composed of the same global optimal policy as the one in sub-graph 2 with the same minimum AoI 2.8809. Finally, the optimal and original distributions of inter-arrival and service time are shown in sub-graphs 3 and 4, respectively.*

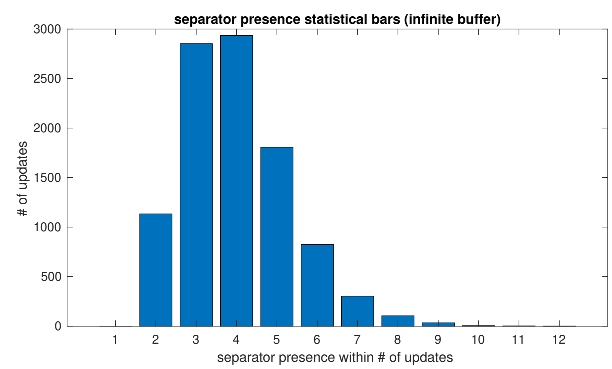

Fig. 5. *Statistical counts of separator presence for a given number N of updates (N = 12) for 10,000 independent experiments, and probability of separator presence increases in less than four updates and decreases from more updates.*

$\lambda = 1.0$, $\mu = 1.0$, the cumulative probability of at least one separator presence within six updates exceeds 90 percentiles.

The curves in Fig.6 demonstrate that the performance of our prediction-based segmentation policy prevails over all of the other optimization policies as arrival rate λ increases. Thus, in comparison with other policies, segmentation policy can find theoretically optimal age of information for a long sequence.

The prediction errors of inter-arrival and service times could be accepted by predictive scheduling policy. The influence on average AoI is illustrated in the Fig.7 and the average AoI decreases as the arrival rate $\lambda \in [3.62, 3.75]$ increases.The performance comparison between different number of predictable

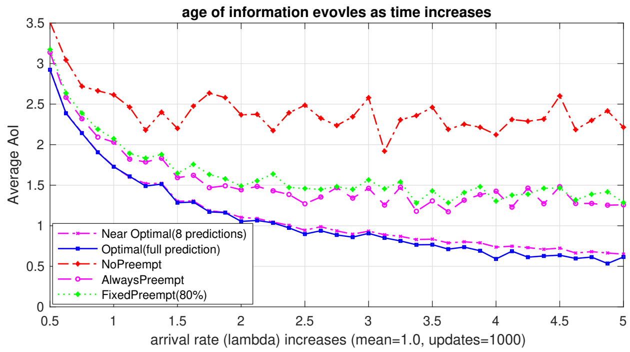

Fig. 6. *Arrival rate λ changes from $\lambda_L = 0.50$ to $\lambda_H = 5.00$ and the mean of service time $\mu = 1.00$. For each λ, 1000 updates are generated and composed of a long sequence. By comparing performance under the same groups of long sequences, this graph shows the difference between various policies.*

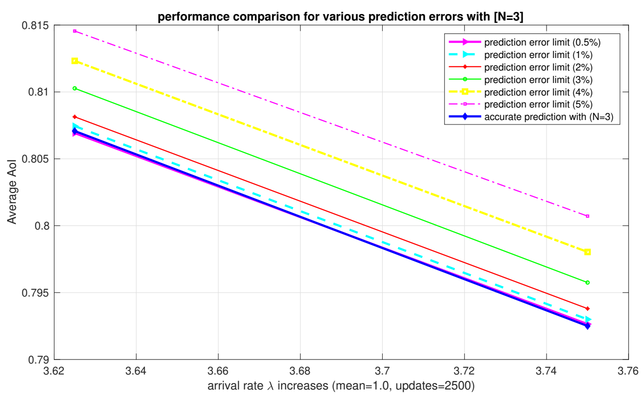

Fig. 7. *The maximum prediction errors of inter-arrival and service times could be possibly accepted by predictive scheduling algorithm, and it still can achieve optimal policy with fault tolerance with errors 0.5%, 1%, 2%, 3%, 4%, 5%.*

packets is shown in the Fig.8. When $N = 4$ number of future packets are predictable, the curve in red color is very close to the optimal curve in blue without estimation error. More details are also given at https://www.overleaf.com/read/yvhwcxrchfjk.

VIII. Conclusion And Future Work

For a status update system with packet sequence predictable [10], this article introduced the segmentation approach and presented the mechanism of discovering epochs. Mathematical expressions for arbitrary epochs are derived, and numerical results are obtained to verify theoretical analysis. Fault tolerance of inaccurate prediction is also considered. The proposed predictive scheduling policy with N number of predictable future packets efficiently reduces the fault requirement of predicting future packets. In future work, more queuing models [9] [12] get explored in extensive analysis and this approach will be applied to scheduling policies in multiple-source settings.

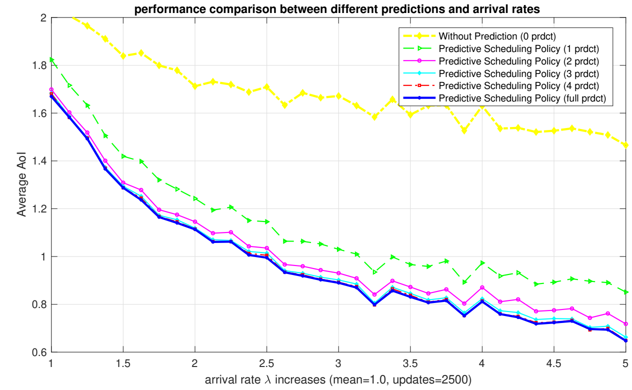

Fig. 8. *N-Prediction Algorithm generates near-optimal AoI under different number of packets $N = 0, 1, 2, 3, 4$. The lower curve in blue color represents AoI under full prediction without estimation error. As the number of predictable future packets increases, average AoI that it is able to achieve decreases rapidly.*

References

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

[2] J. P. Champati, R. R. Avula, T. J. Oechtering and J. Gross. On the minimum achievable age of information for general service-time distributions. In *IEEE INFOCOM*, pages 456-465, 2020.

[3] Sun, Y., Uysal-Biyikoglu, E., Yates, R., Koksal, C. & Shroff, N. Update or wait: How to keep your data fresh. *IEEE Transactions On Information Theory*. **63**, 7492-7508 (2017)

[4] R. Devassy, G. Durisi, G. C. Ferrante, O. Simeone and E. Uysal-Biyikoglu. Delay and Peak-Age Violation Probability in Short-Packet Transmissions. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pp. 2471-2475, 2018.

[5] M. Costa, M. Codreanu and A. Ephremides. On the age of information in status update systems with packet management. In *IEEE Transactions on Information Theory*, 62(4):1897–1910, 2016.

[6] Wang, Q., Chen, H., Gu, Y., Li, Y. & Vucetic, B. Minimizing the age of information of cognitive radio-based IoT systems under a collision constraint. *IEEE Transactions On Wireless Communications*. **19**, 8054-8067 (2020)

[7] Kadota, Igor and Modiano, Eytan Minimizing the Age of Information in Wireless Networks with Stochastic Arrivals. *IEEE Transactions on Mobile Computing*, 2019.

[8] Tong, P., Liu, J., Wang, X., Bai, B. & Dai, H. UAV-enabled age-optimal data collection in wireless sensor networks. *2019 IEEE International Conference On Communications Workshops (ICC Workshops)*. pp. 1-6 (2019)

[9] Champati, J., Al-Zubaidy, H. & Gross, J. On the distribution of AoI for the GI/GI/1/1 and GI/GI/1/2* systems: Exact expressions and bounds. *IEEE INFOCOM 2019-IEEE Conference On Computer Communications*. pp. 37-45 (2019)

[10] Phit, T. & Abe, K. Packet inter-arrival time estimation using neural network models. *Internet Conference, Tokyo*. (2006)

[11] R. D. Yates, Y. Sun, D. R. Brown, S. Kaul, E. Modiano, and S. Ulukus. Age of information: An introduction and survey. *IEEE Journal on Selected Areas in Communications*, 39(5):1183–1210, 2021.

[12] Zou, P. and Zhang, J. and Subramaniam, S. Performance Modeling of Scheduling Algorithms in a Multi-Source Status Update System. *IEEE International Symposium on Information Theory (ISIT)*. pp. 156-161, 2022.

[13] C. Li, S. Li, Y. Chen, Y. T. Hou and W. Lou. AoI scheduling with maximum thresholds. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pages 436–445. IEEE, 2020.

[14] Liu, J., Wang, X., Bai, B. & Dai, H. Age-optimal trajectory planning for UAV-assisted data collection. *IEEE INFOCOM 2018-IEEE Conference On Computer Communications Workshops (INFOCOM WKSHPS)*. pp. 553-558 (2018)

[15] Ayan, O., Vilgelm, M., Klügel, M., Hirche, S. & Kellerer, W. Age-of-information vs. value-of-information scheduling for cellular networked control systems. *Proceedings Of The 10th ACM/IEEE International Conference On Cyber-Physical Systems*. pp. 109-117 (2019)