Minimizing Age-of-Information with Throughput Requirements in Multi-Path Network Communication

Qingyu Liu, Haibo Zeng
Electrical and Computer Engineering
Virginia Tech

Minghua Chen
Information Engineering
The Chinese University of Hong Kong

ABSTRACT

We consider the scenario where a sender periodically sends a batch of data to a receiver over a multi-hop network, possibly using multiple paths. Our objective is to minimize peak/average Age-of-Information (AoI) subject to throughput requirements. The consideration of batch generation and multi-path communication differentiates our AoI study from existing ones. We first show that our AoI minimization problems are NP-hard, but only in the weak sense, as we develop an optimal algorithm with a pseudo-polynomial time complexity. We then prove that minimizing AoI and minimizing maximum delay are "roughly" equivalent, in the sense that any optimal solution of the latter is an approximate solution of the former with bounded optimality loss. We leverage this understanding to design a general approximation framework for our problems. It can build upon any α-approximation algorithm of the maximum delay minimization problem, e.g., the algorithm in [13] with $\alpha = 1 + \epsilon$ given any user-defined $\epsilon > 0$, to construct an $(\alpha + c)$-approximate solution for minimizing AoI. Here c is a constant depending on the throughput requirements. Simulations over various network topologies validate the effectiveness of our approach.

CCS CONCEPTS

• **Mathematics of computing** → **Network flows**; • **Networks** → **Network resources allocation**;

KEYWORDS

Age-of-information, multi-path routing, time-critical network flow

ACM Reference Format:
Qingyu Liu, Haibo Zeng and Minghua Chen. 2019. Minimizing Age-of-Information with Throughput Requirements in Multi-Path Network Communication. In *The Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc '19), July 2–5, 2019, Catania, Italy.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3323679.3326502

1 INTRODUCTION

Age-of-Information (AoI) is a critical networking performance metric for periodic services that require timely transmissions. Kaul *et al.* [7] measures the AoI as the time that elapsed since the last received update was generated. Upon receiving a new packet with updating information, the AoI drops to the elapsed time since the packet generation; otherwise, it grows linearly in time. In this paper, we study fundamental AoI-minimization problems of supporting a periodic transmission task over a multi-hop network. The task requires a sender to send a batch of data (packets) periodically to a receiver, possibly using multiple paths. Our objective is to minimize peak/average AoI subject to both a minimum and a maximum throughput requirement, by jointly optimizing throughput and multi-path routing strategy. We assume the amount of data in the batch is fixed, hence the throughput (the ratio of the volume of the data batch over the task activation period) only varies with the task activation period.

Motivations. Our study is motivated by leveraging a network platform with limited resources to support periodic transmission tasks that are sensitive both to throughputs and to end-to-end delays. A particular example is offloading real-time image-processing tasks in edge computing, with AoI taken into consideration.

Nowadays the blending of mobile/embedded devices and image processing is taking place, where deep learning is often involved to make devices smarter. Since deep learning is resource-heavy, while the mobile/embedded device is resource-constrained, in general those tasks cannot be executed locally on mobile/embedded devices timely as well as frequently. The widely-adopted solution is to leverage nearby powerful edge servers for workload offloading. For example, Ran *et al.* [14] develop an Android application of real-time object detection. If running locally on the phone for 30 minutes, it processes images at a 5 FPS rate and consumes 25% battery. As a comparison, if running remotely on a server, it processes images at the rate of 9 FPS and consumes 15% battery.

From [14] we note that the majority (over 95%) of the total delay of running tasks remotely is the networking delay. Therefore, to offload the resource-heavy image-processing tasks to an edge computing platform for processing in real-time, time-critical offloading algorithms are vital to efficiently and timely utilize available resources. As the results of the offloaded tasks need to be sent to control units, e.g., at the end users or the edge computing nodes, for real-time actions, e.g., cyber-phystical system control, it is important to minimize the age of the information.

We compare our AoI study with existing ones in Tab. 1. Details refer to Sec. 2. In summary, the consideration of batch generation and multi-path communication differentiates our AoI study from existing ones. We study multi-path network communication problems of minimizing peak/average AoI for periodically transmitting a batch of data, subject to both a minimum and a maximum throughput requirement. We claim the following **contributions**.

▷ Comparing minimizing peak/average AoI with minimizing maximum delay: (i) we show that the optimal solution of the former can achieve a throughput that is different from, but always

Mobihoc '19, July 2–5, 2019, Catania, Italy

ACM ISBN 978-1-4503-6764-6/19/07…$15.00
https://doi.org/10.1145/3323679.3326502

Table 1: Compare our AoI study with existing ones.

		[7, 8, 17]	[16]	[4]	[20]	[6]	[18, 19]	[21]	Our work
Objective of Optimization	Minimize peak AoI	✗	✓	✓	✓	✗	✓	✓	✓
	Minimize average AoI	✓	✓	✗	✓	✓	✓	✓	✓
Design Space of Optimization	Multi-path routing strategy	✗	✗	✗	✗	✗	✗	✗	✓
	Information generation rate*	✓	✗	✓	✓	✓	✗	✓	✓
	Link scheduling policy	✗	✗	✗	✗	✓	✓	✓	✗
	Queuing disciplines	✗	✓	✓	✓	✗	✗	✗	✗
Other Results	Compare AoI with delay	✓	✗	✓	✓	✗	✗	✗	✓

Note. *: Under our system model, the information generation rate is equivalent to the achieved throughput.

no smaller than, that achieved by the optimal solution of the latter (Lem. 4.2). This result is consistent with our observation that AoI is a metric simultaneously considering maximum delay and throughput; (ii) we show that the optimal solution of the latter can be suboptimal to the former (Lem. 4.2), but with a bounded optimality loss (Lem. 4.3).

▷ Comparing minimizing peak AoI with minimizing average AoI: (i) we prove that the optimal solution of the former can be suboptimal to the latter, and vice versa, but both with bounded optimality losses (Lem. 5.1); (ii) we show that the optimal solution of the former can achieve a throughput (resp. maximum delay) that is different from, but always no smaller than, that achieved by the optimal solution of the latter (Lem. 5.1). Thus, the problem of minimizing peak AoI may carry more flavor on throughput and less on maximum delay, compared to that of minimizing average AoI.

▷ We observe that both minimizing peak AoI and minimizing average AoI are challenging, because (i) we prove that both minimal peak AoI and minimal average AoI are non-monotonic, non-convex, and non-concave with throughput theoretically (Lem. 5.5), and (ii) we prove that both problems are NP-hard (Lem. 5.2), but in the weak sense (Thm. 5.6), as we design an algorithm to solve them optimally in a pseudo-polynomial time (Sec. 5.4).

▷ We further leverage our understanding on comparing AoI with maximum delay to develop an approximation framework (Thm. 6.2). It can build upon any α-approximation algorithm of the maximum delay minimization problem, e.g., the algorithm in [13] with $\alpha = 1+\epsilon$ given any user-defined $\epsilon > 0$, to construct an $(\alpha + c)$-approximate solution for minimizing AoI. Here c is a constant depending on the throughput requirements. Our framework has the same time complexity as that of the used α-approximation algorithm, and suggests a new avenue for designing approximation algorithms for minimizing AoI in the field of multi-path network communication.

▷ We conduct extensive simulations to evaluate our proposed approaches (Sec. 7). Empirically (i) our optimal algorithm obtains more than 3% AoI reduction compared to our approximation framework, if the range of task activation period increases by 1. However, (ii) our approximation framework has a constant running time of 0.06s, while the running time of our optimal algorithm can increase by 0.12s if the range of task activation period increases by 1.

2 RELATED WORK

Since introduced by [7], AoI has been studied theoretically and experimentally by various studies, which are summarized in Tab. 1. We differ from existing AoI studies in two aspects, i.e., the problem design space and the AoI definition.

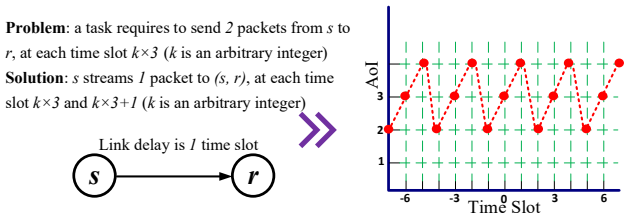


Figure 1: An illustrative example of our batch-based AoI. Sender s generates a batch of two packets at each slot $3k, \forall k \in \mathbb{Z}$. It sends the two packets one-by-one over link (s, r) to the receiver r; the link transmission incurs one-slot delay. Our batch-based AoI drops only when all packets from the same batch are received by r. Hence, as receiver r receives all the two packets in a batch at each slot $3k + 2, \forall k \in \mathbb{Z}$, the batch-based AoI at each slot $3k + 2$ drops to 2, i.e., the elapsed time since the generation of the last received batch. The batch-based AoI at all the other slot grows linearly.

We note that multi-path routing is a basic paradigm of network communication. It is a natural extension of the single-path routing when streaming a high volume of traffic while avoid link traffic congestions. Many existing studies, e.g., [22–24], have shown that multi-path routing can provide better QoS, e.g., larger throughput, than the single-path routing. To our best knowledge, this is the first work to optimize the multi-path routing strategy to minimize AoI.

Besides, existing studies define AoI at the packet level. Such definitions assume that AoI can be updated by receiving any packet, which are reasonable in status update systems. However, in our task-level study, we fairly assume that the receiver can reconstruct information of one task period and hence update AoI accordingly, only after it receives all the packets in a batch belonging to that task period. By this assumption, our batch-based AoI drop only upon successful reception of the complete batch of data belonging to one task period, and increase linearly otherwise. We give an illustrative example in Fig. 1, assuming slotted data transmissions.

Note that our problems are challenging, further compared to existing time-critical multi-path communication studies. Our problems minimize AoI with throughput requirements. Thus the task activation period (the ratio of the amount of data in the batch over throughput) is a decision variable under our setting. In contrast, to our best knowledge, existing time-critical multi-path communication problems minimize maximum delay given a fixed task activation period. Such problems include the quickest flow problem

and the min-max-delay flow problem. Here the maximum delay is the time of sending a complete batch of data from the sender to the receiver, and clearly that AoI is a metric jointly considering maximum delay and task activation period.

Quickest flow problem [10, 15]. Given an amount of data, it finds the minimum time needed to send them from a sender to a receiver, and the corresponding multi-path routing solution. This problem assumes that the task activation period is infinitely large, and is polynomial-time solvable under our setting [10].

Min-max-delay flow problem [11, 13]. Given a sender-receiver pair and an amount of data, it finds a set of sender-to-receiver paths such that the maximum path delay of the set of paths is minimized while the aggregate bandwidth of the set of paths is no smaller than the given amount of data. This problem (also known as problem OMPBD studied in [13]) assumes that the task activation period is one unit of time, and is NP-hard under our setting [13].

If D is the given amount of data, note again that for the task activation period T, under our setting, we have $T = +\infty$ for the quickest flow problem, $T = 1$ for the min-max-delay flow problem, but $D/R_u \leq T \leq D/R_l$ for our problems where R_u (resp. R_l) is our maximum (resp. minimum) throughput requirement. Thus for the throughput R, we have $R \to 0$ for the quickest flow problem, $R = D$ for the min-max-delay flow problem, but $0 < R_l \leq R \leq R_u \leq D$ for our problems. According to Lem. 4.1 introduced later, given $R_l = R_u$, minimizing AoI is equivalent to minimizing maximum delay. This implies that the quickest flow problem ($R_l = R_u \to 0$) and the min-max-delay flow problem ($R_l = R_u = D$) are special cases of our problems. However, although exact algorithms for the quickest flow problem [10] and approximation algorithms for the min-max-delay flow problem [13] have been developed, it is still not clear how to solve our problems even given $0 < R_l = R_u \leq D$. Moreover, according to Lem. 4.2, given $0 < R_l < R_u \leq D$, minimizing AoI can differ from minimizing maximum delay. Overall, we observe that our AoI-minimization problems are uniquely challenging.

In the literature there exist some other time-critical periodic communication studies. For example, Hou et al. [3] propose scheduling policies for a set of senders to be feasible with respect to the delay constraint, throughput constraint, and wireless channel reliability constraint. Deng et al. [2] further conduct a complete study on the similar timely wireless flow problem but assuming a more general traffic pattern. Those studies [2, 3] are of little relevance with our problems, because their focus is the wireless link scheduling policy optimization. Differently, we focus on the throughput optimization and the multi-path routing strategy optimization.

3 SYSTEM MODEL

3.1 Preliminary

We consider a multi-hop network modeled as a directed graph $G \triangleq (V, E)$ with $|V|$ nodes and $|E|$ links. We assume slotted data transmissions. Each link $e \in E$ has a bandwidth b_e and a delay d_e. At the beginning of each time slot, each link e can stream an amount of data that is no larger than the bandwidth $b_e \in \mathbb{R}, b_e \geq 0$ (b_e is a non-negative real number) to it, and this data experiences a delay of $d_e \in \mathbb{Z}^+$ (d_e is a positive integer) slots to pass it. Besides, we assume that each node $v \in V$ can hold an arbitrary amount of data at each time slot. For easier reference, in this paper, we use "at time t" to refer to "at the beginning of the time slot t". We focus on a task that requires a periodic data transmission. Specifically, given that the **task activation period** is $T \in \mathbb{Z}^+$, the task will generate $D \in \mathbb{R}, D > 0$ amount of data at a sender node $s \in V$ at time kT for each $k \in \mathbb{Z}$ (k is an integer), and is required to transmit them to a receiver node $r \in V \backslash \{s\}$, possibly using multiple paths. Because we assume no data loss during transmission, the throughput incurred by a task activation period of T is D/T.

Table 2: Summary of important notations.

$\mathcal{T}(f)$	Task activation period of a solution f
$\Lambda_p(f)$ (resp. $\Lambda_a(f)$, $\mathcal{M}(f)$)	Peak AoI (resp. Average AoI, Maximum delay) of f
R_l (resp. R_u)	Input minimum (resp. maximum) throughput requirement
Λ_p^R (resp. Λ_a^R, $\mathcal{M}^R$)	Minimal peak AoI (resp. Minimal average AoI, Minimal maximum delay) that can be achieved by any feasible periodically repeated solution with a throughput of R
R_p (resp. R_a, R_m)	Achieved throughput that is optimal to our peak AoI (resp. average AoI, maximum delay) minimization problem

We aim to obtain a "fresh" multi-path routing solution that is *periodically repeated* to periodically send the batch of data. Here the "freshness" is evaluated by AoI that is a function of the end-to-end networking delay (see our formula (4)). It is well-known that the networking delay is mainly composed of propagation delay, transmission delay, and queuing delay. Similar to the discussions in [1], we remark that the slotted data transmission model can take all different kinds of delays into consideration (see Appendix 9.1 of our technical report [12]). We denote the set of all simple paths from s to r as P. For a path $p \in P$, we denote the number of nodes belonging to p as $|p|$. There are different ways to describe a periodically repeated solution f, one of which defines f as the assigned amount of data over P at the time offset $\vec{\mathcal{U}}$,

$$f \triangleq \left\{x^p(\vec{u}) \geq 0 : \ \forall p \in P, \forall \vec{u} \in \vec{\mathcal{U}}\right\}, \tag{1}$$

where $\vec{\mathcal{U}}$ is defined as follows: suppose $p \in P$ is an arbitrary path and $p = \langle v_1, v_2, ..., v_{|p|}\rangle$, where $\{v_i \in V, i = 1, 2, ..., |p|\}$ are the nodes on p and $\{e_{i-1} = (v_{i-1}, v_i) \in E, i = 2, 3, ..., |p|\}$ are the links belonging to p, with $v_1 = s$ and $v_{|p|} = r$. Any offset $\vec{u} \in \vec{\mathcal{U}}$ corresponding to the path p is described by $\vec{u} = \langle u_0, u_1, u_2, ..., u_{|p|}\rangle$, with the following held assuming $u_0 = 0$ and $d_{e_0} = 0$

$$u_i \in \mathbb{Z} \text{ and } u_i \in [u_{i-1}+d_{e_{i-1}}, u_{i-1}+d_{e_{i-1}}+U], \ \forall i = 1, 2, ..., |p|. \tag{2}$$

Each positive $x^p(\vec{u})$ of f requires us to push $x^p(\vec{u})$ amount of data onto link (v_i, v_{i+1}) at the offset u_i, i.e., push $x^p(\vec{u})$ amount of data of the period that starts at time $k \cdot \mathcal{T}(f)$ onto link (v_i, v_{i+1}) at time $k \cdot \mathcal{T}(f) + u_i$, where $\mathcal{T}(f)$ is the task activation period of f. We remark that in the definition (2), we have $u_i - d_{e_{i-1}} - u_{i-1} \leq U, \forall i = 1, 2, ..., |p|$. This is equivalent to restricting the data-holding delay of each node to be no more than U slots. Because in this paper we assume each node can hold an arbitrary amount of data at each slot, for our problems $U = +\infty$. However, as proved later in Lem. 3.1, setting $U = T-1$ is large enough for us to solve any feasible instance of our problems, if we are interested in solutions that have a task activation period of T. Overall, each positive $x^p(\vec{u})$ of f requires us

to transmit $x^p(\vec{u})$ amount of data in a batch from s to r, following the path p and the time offset $\vec{u}$.

Given a solution f, based on each positive $x^p(\vec{u})$ of f, we can easily figure out (i) the beginning offset of pushing those data onto the link $e_i \in p, i = 1, ..., |p| - 1$, denoted as $\mathcal{B}^p(\vec{u}, e_i)$,

$$\mathcal{B}^p(\vec{u}, e_i) \;=\; u_i,$$

and (ii) the end-to-end delay for those data to travel from s to r, denoted as $\mathcal{A}^p(\vec{u})$,

$$\mathcal{A}^p(\vec{u}) \;=\; \mathcal{B}^p(\vec{u}, e_{|p|-1}) + d_{e_{|p|-1}} \;=\; u_{|p|-1} + d_{e_{|p|-1}}.$$

One important time-aware networking performance metric of f is the **maximum delay**, denoted as $\mathcal{M}(f)$. It is the time difference comparing the time when the batch of data of one period is received by the receiver r, to the beginning time of this period when those data is generated at the sender s waiting for transmission, i.e.,

$$\mathcal{M}(f) \;\triangleq\; \max_{\forall p \in P, \forall \vec{u} \in \vec{\mathcal{U}}:\, x^p(\vec{u}) > 0} \mathcal{A}^p(\vec{u}). \tag{3}$$

In order to measure the time that elapsed since the generation of the task period that was most recently delivered to the receiver, we define the **AoI** of f at time t, denoted by $\mathcal{I}(f, t)$, as

$$\mathcal{I}(f, t) \;\triangleq\; t - \pi_t(f), \tag{4}$$

where $\pi_t(f)$ is the generation time of the task period that was most recently delivered to r by time t, i.e.,

$$\pi_t(f) \;\triangleq\; \max_{k \in \mathbb{Z}} \{k \cdot \mathcal{T}(f) :\; k \cdot \mathcal{T}(f) + \mathcal{M}(f) \le t\}.$$

3.2 Problem Definition

In this paper we focus on the minimization of (i) the peak value of AoI, and (ii) the average value of AoI, both over all the time slots. We define the **peak AoI** of f, denoted as $\Lambda_p(f)$, as follows

$$\Lambda_p(f) \;\triangleq\; \max_{t \in \mathbb{Z}} \mathcal{I}(f, t), \tag{5}$$

and define the **average AoI** of f, denoted as $\Lambda_a(f)$, as

$$\Lambda_a(f) \;\triangleq\; \sum_{t \in \mathbb{Z}} \mathcal{I}(f, t) \,/\, \sum_{t \in \mathbb{Z}} 1. \tag{6}$$

Our problems of finding a periodically repeated solution f to minimize AoI are subject to a minimum throughput requirement, a maximum throughput requirement, and link bandwidth constraints. The minimum (resp. maximum) **throughput requirement** requires f to send D amount of data every $\mathcal{T}(f) \in \mathbb{Z}^+$ time slots, achieving a throughput no smaller than an input $R_l \in \mathbb{R}$ (resp. no greater than an input $R_u \in \mathbb{R}$), i.e.,

$$\sum_{\forall p \in P} \sum_{\forall \vec{u} \in \vec{\mathcal{U}}} x^p(\vec{u}) = D, R_l \le D/\mathcal{T}(f) \le R_u, \text{ and } \mathcal{T}(f) \in \mathbb{Z}^+. \tag{7}$$

It is clear for us to fairly assume $D/R_l \in \mathbb{Z}^+$ and $D/R_u \in \mathbb{Z}^+$ for the input R_l and R_u, due to $\mathcal{T}(f) \in \mathbb{Z}^+$.

Given a solution f, we denote the aggregate amount of data sent to link $e \in E$ at the offset $i \in \{0, 1, ..., \mathcal{T}(f) - 1\}$, or equivalently the aggregate amount of data sent to e at each time $k \cdot \mathcal{T}(f) + i, \forall k \in \mathbb{Z}$, as $x_e(i)$. Note that $x_e(i)$ may include data assigned to different path-offset pairs of one period, and may even include data from multiple periods with different starting times. We remark that $0 \le i \le \mathcal{T}(f) - 1, i \in \mathbb{Z}$, because $x_e(i + \mathcal{T}(f))$ is always equal to $x_e(i)$ considering that f is periodically repeated. Specifically, (i) $x_e(i + \mathcal{T}(f))$ is the aggregate data assigned to e at the offset $i + \mathcal{T}(f)$, i.e., at time $k \cdot \mathcal{T}(f) + i + \mathcal{T}(f)$ from the perspective of the period starting at time $k \cdot \mathcal{T}(f)$, and (ii) $x_e(i)$ is the aggregate data assigned to e at the offset i, i.e., also at time $k \cdot \mathcal{T}(f) + i + \mathcal{T}(f)$ but from the perspective of the period starting at time $(k + 1) \cdot \mathcal{T}(f)$. The link bandwidth constraints require $x_e(i)$ to be no greater than b_e, i.e., $x_e(i) \le b_e$, for any link $e \in E$ and any offset $i = 0, 1, ..., \mathcal{T}(f) - 1$. This is equivalent to restricting that the aggregate data sent to each link $e \in E$ at each time slot shall be upper bounded by b_e.

It is clear that $x^p(\vec{u})$ will contribute to $x_e(i)$ if and only if $e \in p$ and there exists a $k \in \mathbb{Z}$ such that $k \cdot \mathcal{T}(f) + \mathcal{B}^p(\vec{u}, e) = i$. Therefore, our **link bandwidth constraints** are equivalent to the following

$$\sum_{\substack{p \in P:\\ e \in p}} \sum_{\substack{k \in \mathbb{Z}, \vec{u}: k \cdot \mathcal{T}(f) \\ + \mathcal{B}^p(\vec{u}, e) = i}} x^p(\vec{u}) \;\le\; b_e,\; \forall i = 0, ..., \mathcal{T}(f) - 1, \forall e \in E. \tag{8}$$

Suppose Λ_p^R (resp. Λ_a^R) is the minimal peak AoI (resp. minimal average AoI) that can be achieved by any periodically repeated solution which obtains a throughput of R, meeting link bandwidth constraints. Now given a network $G(V, E)$, a sender $s \in V$, a receiver $r \in V \backslash \{s\}$, throughput requirements R_l and R_u, in this paper we are interested in the following two AoI minimization problems,

(1) Obtain an optimal throughput $R_p \in [R_l, R_u], D/R_p \in \mathbb{Z}^+$ that achieves the minimal peak AoI, i.e.,

$$R_p \;\triangleq\; \arg \min_{R_l \le R \le R_u, D/R \in \mathbb{Z}^+} \Lambda_p^R.$$

and obtain the feasible periodically repeated solution which has a throughput of R_p and a peak AoI of $\Lambda_p^{R_p}$. We denote this problem of Minimizing Peak AoI as **MPA**.

(2) Obtain an optimal throughput $R_a \in [R_l, R_u], D/R_a \in \mathbb{Z}^+$ that achieves the minimal average AoI, i.e.,

$$R_a \;\triangleq\; \arg \min_{R_l \le R \le R_u, D/R \in \mathbb{Z}^+} \Lambda_a^R,$$

and obtain the feasible periodically repeated solution which has a throughput of R_a and an average AoI of $\Lambda_a^{R_a}$. We denote this problem of Minimizing Average AoI as **MAA**.

As discussed in Sec. 2, existing time-critical multi-path communication problems minimize maximum delay, instead of AoI. Similar to MPA and MAA, we can define (i) $\mathcal{M}^R$ as the minimal maximum delay with a throughput of R, and (ii) problem of Minimizing Maximum Delay (**MMD**) as the problem of obtaining an optimal $R_m \in [R_l, R_u], D/R_m \in \mathbb{Z}^+$ that achieves minimal maximum delay, and obtaining associated optimal periodically repeated solution.

Finally, we give one lemma which argues for any feasible solution g whose data-holding delay may exceed $\mathcal{T}(g) - 1$ slots for certain node, there must exist a feasible solution f whose data-holding delay is no more than $\mathcal{T}(f) - 1$ slots for all nodes, and the following holds comparing g with f: (i) they achieve the same throughput, and (ii) the peak AoI (resp. average AoI) of f is no worse than that of g. A direct corollary is for any feasible MPA (resp. MAA) instance, setting U (see formula (1)) to be $T - 1$ is large enough for us to solve it, if we are interested in solutions which have a task activation period of T and thus a throughput of D/T.

Lemma 3.1. *Given any instance of MPA (or MAA), suppose g is an arbitrary feasible periodically repeated solution. Then there must exist another feasible periodically repeated solution f, where $\mathcal{T}(f) = \mathcal{T}(g)$, $\Lambda_p(f) \le \Lambda_p(g)$, $\Lambda_a(f) \le \Lambda_a(g)$, and for each positive $x^p(\vec{u})$ (suppose $p = \langle v_1, ..., v_{|p|} \rangle$ and $\vec{u} = \langle u_0, ..., u_{|p|} \rangle$) of f, we have $u_i - d_{e_{i-1}} - u_{i-1} \le \mathcal{T}(f) - 1$, for all $i = 1, 2, ..., |p|$.*

Proof. Refer to Appendix 9.2 of our technical report [12] □

4 COMPARE AOI WITH MAXIMUM DELAY

As time-critical networking performance metrics, maximum delay is well-known, while AoI is newly proposed. In this section, we compare the problem of minimizing AoI (MPA and MAA) with that of minimizing maximum delay (MMD) theoretically.

Consider the following example. In a network with nodes s and r, and one link (s, r). Suppose the delay (resp. bandwidth) of the link is d (resp. $b \ge D$). Suppose $R_u = D$ and $R_l = D/T_u$ given a $T_u \in \mathbb{Z}^+$. Consider one solution that streams D data to (s, r) at the offset 0. It is clear that this solution can have a task activation period of $T \le T_u$, meeting throughput requirements and link bandwidth constraints. And the batch of data of the period starting at time kT will be received by r at time $kT + d$. Now consider two different task activation periods T_1 and T_2 with $T_1 < T_2 \le T_u$. From the perspective of minimizing maximum delay, the solution with $T = T_1$ is equivalent to that with $T = T_2$, because they are both feasible, and obtain the same maximum delay of d. From the perspective of minimizing peak/average AoI, in contrast, the solution with $T = T_1$ is better than that with $T = T_2$, since according to Lem. 4.1 introduced later, the peak AoI (resp. average AoI) of former is $d + T_1 - 1$ (resp. $d + (T_1 - 1)/2$), which is smaller than that of latter, i.e., than $d + T_2 - 1$ (resp. $d + (T_2 - 1)/2$). In fact, T_1 is better than T_2 in this example, because they lead to the same delay of periodically transmitting the batch of data, but the throughput achieved by T_1 (D/T_1) is greater than that achieved by T_2 (D/T_2).

For periodic transmission services, AoI, instead of maximum delay, should be optimized to provide time-critical solutions according to the example. This is mainly because AoI is a time-critical metric simultaneously considering throughput and maximum delay. In the following, we further prove that the maximum-delay-optimal solution can achieve a suboptimal peak/average AoI, but it must be with bounded optimality loss compared to optimal.

Given a solution f, first we give a lemma to mathematically relates the peak/average AoI of f to the maximum delay of f.

Lemma 4.1. *For an arbitrary periodically repeated solution f, we have the following*

$$\Lambda_p(f) = \mathcal{M}(f) + \mathcal{T}(f) - 1,\ \Lambda_a(f) = \mathcal{M}(f) + (\mathcal{T}(f) - 1)/2.$$

Proof. Refer to Appendix 9.3 of our technical report [12] □

A direct corollary is that the peak AoI (resp. average AoI) of a feasible solution which achieves a throughput of R and has a maximum delay of $\mathcal{M}^R$ is Λ_p^R (resp. Λ_a^R). Thus to solve MPA and MAA given $R_l = R_u$, we can solve the corresponding MMD instead. However, as introduced in Sec. 2, only special cases of MMD with $R_l = R_u$, i.e., the quickest flow problem ($R_l = R_u \to 0$) and the min-max-delay flow problem ($R_l = R_u = D$), are studied in the literature, and it is not clear how to solve MMD even given $0 < R_l = R_u \le D$. Moreover, for general settings with $R_l < R_u$, we observe that both MPA and MAA can differ from MMD as follows.

Lemma 4.2. *Given any instance of MPA (or MAA, MMD), suppose $\vec{R}_p$ (resp. $\vec{R}_a$, $\vec{R}_m$) is the optimal set of throughputs that minimize peak AoI (resp. average AoI, maximum delay) of this instance. The following must hold for this instance*

$$\min_{R_p \in \vec{R}_p} R_p \ge \max_{R_m \in \vec{R}_m} R_m,\ \min_{R_a \in \vec{R}_a} R_a \ge \max_{R_m \in \vec{R}_m} R_m.$$

And there must exist an instance where the following holds

$$\min_{R_p \in \vec{R}_p} R_p > \max_{R_m \in \vec{R}_m} R_m,\ \min_{R_a \in \vec{R}_a} R_a > \max_{R_m \in \vec{R}_m} R_m.$$

Proof. Refer to Appendix 9.4 of our technical report [12] □

In Lem. 4.2, $\vec{R}_p$ is defined as a set of throughputs, because in certain instances there may exist multiple throughputs obtaining the same and optimal peak AoI. Similarly, we define $\vec{R}_a$ and $\vec{R}_m$ both as sets of throughputs.

Lem. 4.2 suggests that (i) minimizing maximum delay can differ from minimizing AoI, because the maximum-delay-optimal solution can achieve suboptimal peak/average AoI. (ii) The throughput of the maximum-delay-optimal solution must be no greater than that of the peak-/average- AoI-optimal solution. In the following, we further characterize near-tight optimality losses for the suboptimal AoI achieved by the maximum-delay-optimal solution.

Lemma 4.3. *Given any instance of MPA (or MAA, MMD), suppose $\vec{R}_p$ (resp. $\vec{R}_a$, $\vec{R}_m$) is the optimal set of throughputs that minimize peak AoI (resp. average AoI, maximum delay) of this instance. The following must hold for this instance*

$$\Lambda_p^{R_m} - \Lambda_p^{R_p} \le \frac{D}{R_l} - \frac{D}{R_u}, \forall R_m \in \vec{R}_m, \forall R_p \in \vec{R}_p. \quad (9)$$

$$\Lambda_a^{R_m} - \Lambda_a^{R_a} \le \frac{D}{2R_l} - \frac{D}{2R_u}, \forall R_m \in \vec{R}_m, \forall R_a \in \vec{R}_a. \quad (10)$$

Gap (9) *is near-tight, in the sense that for arbitrary D, R_l, and R_u that meet $D > 0$, $D/R_l \in \mathbb{Z}^+$, and $D/R_u \in \mathbb{Z}^+$, there is an instance where the following holds*

$$\Lambda_p^{R_m} - \Lambda_p^{R_p} \ge \frac{D}{R_l} - \frac{D}{R_u} - 1,\ \forall R_m \in \vec{R}_m, \forall R_p \in \vec{R}_p.$$

Gap (10) *is near-tight, in a similar sense with the following held*

$$\Lambda_a^{R_m} - \Lambda_a^{R_a} \ge \frac{D}{2R_l} - \frac{D}{2R_u} - 1,\ \forall R_m \in \vec{R}_m, \forall R_a \in \vec{R}_a.$$

Proof. Refer to Appendix 9.5 of our technical report [12] □

Overall, we observe that MPA and MAA are non-trivial as compared to MMD: (i) AoI-optimal solution, instead of maximum-delay-optimal one, is the time-critical solution for periodic transmission

services; (ii) AoI-optimal solution can differ from the maximum-delay-optimal one in the general scenario with throughput optimization involved ($R_l < R_u$); (iii) even for the special scenario where the throughput of feasible solutions is fixed ($R_l = R_u$), where it can be proved that the AoI-optimal solution is also maximum-delay-optimal, and vice versa, existing maximum delay minimization studies have strong assumptions on the fixed throughput (either $R_l = R_u \to 0$ or $R_l = R_u = D$), and it is not clear how to minimize maximum delay with the throughput fixed arbitrarily ($0 < R_l = R_u \le D$). In the following sections, we design an optimal algorithm and an approximation framework for MPA and MAA.

5 PROBLEM STRUCTURES OF MPA AND MAA

In this section we give a complete understanding on the fundamental structures of our MPA and MAA. In particular, we first show that MPA and MAA are two different problems theoretically, and then prove that they are both NP-hard in the weak sense, with a pseudo-polynomial-time optimal algorithm developed.

5.1 MPA is Different from MAA

Comparing MPA of minimizing peak AoI with MAA of minimizing average AoI, we observe that they are two different problems, as proved in the following lemma.

Lemma 5.1. *Given any instance of MPA (or MAA), suppose $\vec{R}_p$ (resp. $\vec{R}_a$) is the optimal set of throughputs that minimize peak AoI (resp. average AoI) of this instance. For this instance,*

(1) *the following must hold*

$$\min_{R_p \in \vec{R}_p} R_p \ge \max_{R_a \in \vec{R}_a} R_a, \quad \min_{R_p \in \vec{R}_p} \mathcal{M}^{R_p} \ge \max_{R_a \in \vec{R}_a} \mathcal{M}^{R_a},$$

(2) *and we have the following*

$$\Lambda_a^{R_p} - \Lambda_a^{R_a} \le \frac{D}{2R_l} - \frac{D}{2R_u}, \forall R_p \in \vec{R}_p, \forall R_a \in \vec{R}_a. \quad (11)$$

$$\Lambda_p^{R_a} - \Lambda_p^{R_p} \le \left\lfloor \frac{D}{2R_l} - \frac{D}{2R_u} \right\rfloor, \forall R_p \in \vec{R}_p, \forall R_a \in \vec{R}_a. (12)$$

Moreover, there must exist an instance where the following holds

$$\min_{R_p \in \vec{R}_p} R_p > \max_{R_a \in \vec{R}_a} R_a, \quad \min_{R_p \in \vec{R}_p} \mathcal{M}^{R_p} > \max_{R_a \in \vec{R}_a} \mathcal{M}^{R_a},$$

Proof. Refer to Appendix 9.6 of our technical report [12] □

From the lemma, we learn that (i) MPA can differ from MAA, and (ii) although both MPA and MAA minimize AoI which jointly considers throughput and maximum delay, we observe that MPA of minimizing peak AoI may carry more flavor on throughput and less on maximum delay, compared to MAA of minimizing average AoI. In the lemma, (iii) we further characterize bounded optimality loss for the suboptimal average AoI (resp. suboptimal peak AoI) achieved by the optimal solution to MPA (resp. to MAA).

5.2 MPA and MAA are both NP-Hard

Although MPA differs from MAA, we observe that they are both NP-hard, because (i) based on Lem. 4.1, MMD given $R_l = R_u = D$ is a special case of MPA and MAA. (ii) As discussed in Sec. 2, the min-max-delay flow problem under our setting is exactly the problem MMD given $R_l = R_u = D$, and it has been proven to be NP-hard by the study [13]. Overall, we have the following.

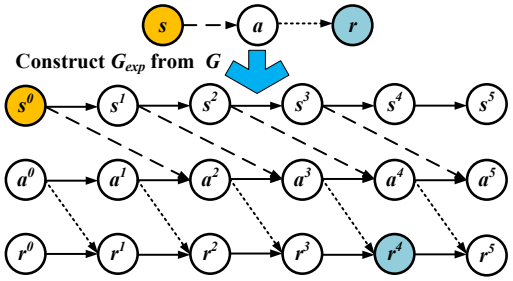


Figure 2: An example of constructing $G_{\exp}$. Suppose $d_{(s,a)} = 2$ and $d_{(a,r)} = 1$ in G. And suppose $M_U = 5$ and $M = 4$.

Lemma 5.2. *MPA and MAA are both NP-hard.*

Proof. Both MPA and MAA cover the NP-hard min-max-delay flow problem [13] as a special case. □

In the following we propose a pseudo-polynomial-time algorithm which solves MPA (resp. MAA) optimally. It enumerates all possible throughputs $R \in [R_l, R_u], D/R \in \mathbb{Z}^+$ to figure out the peak-AoI-optimal R_p (resp. average-AoI-optimal R_a), together with the optimal periodically repeated solution.

5.3 Design an Algorithm 2 to Obtain $\mathcal{M}^R$ in a Pseudo-Polynomial Time

Given a throughput R with $D/R \in \mathbb{Z}^+$, first we design a pseudo-polynomial-time algorithm which leverages a binary-search based scheme, together with an expanded network, to figure out the minimal maximum delay $\mathcal{M}^R$ and the corresponding solution. According to Lem. 4.1, the minimal peak AoI Λ_p^R and the minimal average AoI Λ_a^R can be achieved by the same solution.

Construct an expanded network. We construct an expanded network $G_{\exp}(V_{\exp}, E_{\exp})$, from the input $G(V, E)$ following Algorithm 1. Given an integer M_U that is an upper bound of $\mathcal{M}^R$, first we expand each node $v \in V$ to nodes $v^i, i = 0, ..., M_U$ (the loop in line 5). By this expansion, node v^i represents the node v at time $kT + i$ from the perspective of the period starting at time $kT, \forall k \in \mathbb{Z}$, where $T = D/R$. Second we expand each link $e = (v, w) \in E$ to links $(v^i, w^{i+d_e}), i = 0, ..., M_U - d_e$ (the loop in line 7). By this expansion, the link (v^i, w^{i+d_e}) represents that certain amount of data can be streamed to the link (v, w) at time $kT + i$ from the perspective of the period starting at time $kT, \forall k \in \mathbb{Z}$, where $T = D/R$. Third, we add links $(v^i, v^{i+1}), i = 0, ..., M_U - 1$ (the loop in line 9) for each $v \in V$, because we allow each node to hold data at each time slot.

Obtain $\mathcal{M}^R$ using binary search. Given an arbitrary integer M with $M \le M_U$, we observe that the problem of whether there exists a feasible periodically repeated solution f in G, with $\mathcal{T}(f) = D/R$ and $\mathcal{M}(f) \le M$, can be solved by solving a network flow problem

Algorithm 1 Construct G_{exp} from G

input: $G = (V, E)$, M_U
output: $G_{\text{exp}} = (V_{\text{exp}}, E_{\text{exp}})$
procedure
  $V_{\text{exp}} = E_{\text{exp}} = \text{NULL}$
  for $v \in V$ and $i = 0, 1, ..., M_U$ **do**
    Push node v^i into V_{exp}
  for $e = (v, w) \in E$ and $i = 0, 1, ..., M_U - d_e$ **do**
    Push link (v^i, w^{i+d_e}) into E_{exp}
  for $v \in V$ and $i = 0, 1, ..., M_U - 1$ **do**
    Push link (v^i, v^{i+1}) into E_{exp}
  return $G_{\text{exp}} = (V_{\text{exp}}, E_{\text{exp}})$

that is casted by the following linear program in G_{exp}.

$$\max \quad \sum_{e' \in \text{Out}(s^0)} x_{e'} \tag{13a}$$

$$\text{s.t.} \quad \sum_{e' \in \text{Out}(s^0)} x_{e'} = \sum_{e' \in \text{In}(r^M)} x_{e'}, \tag{13b}$$

$$\sum_{e' \in \text{Out}(v)} x_{e'} = \sum_{e' \in \text{In}(v)} x_{e'}, \forall v \in V_{\text{exp}} \backslash \{s^0, r^M\}, \tag{13c}$$

$$\sum_{e' \in e(i)} x_{e'} \le b_e, \ \forall e \in E, \forall i = 0, 1, ..., D/R - 1, \tag{13d}$$

$$\text{vars.} \quad x_{e'} \ge 0, \ \forall e' \in E_{\text{exp}}. \tag{13e}$$

Here $\text{In}(v)$ (resp. $\text{Out}(v)$) is the set of incoming (resp. outgoing) links of $v \in V_{\text{exp}}$ in G_{exp}. Suppose $e = (v, w) \in E$, then $e(i)$ is the set of expanded links $\{(v^{kT+i}, w^{kT+i+d_e}), \forall k \in \mathbb{Z}\}$ that belong to E_{exp}, where $T = D/R$. Note that data assigned to $e(i)$ must aggregately respect bandwidth constraint of b_e, considering that the aggregate data assigned to $e(i)$ is exactly equal to $x_e(i)$ that is introduced in the definition of our link bandwidth constraints (8). This is because the difference of starting times of links belonging to $e(i)$ are multiples of the task activation period D/R. The objective (13a) maximizes the amount of data sent from the sender of each period. Constraint (13b) restricts those data arrive at the receiver no later than M time slots as compared to the starting time of the period. Constraints (13c) are flow conservation constraints, and constraints (13d) are link bandwidth constraints. We remark again that the constraints (13d) restricts that the aggregate data pushed onto e at each time $kT + i, \forall k \in \mathbb{Z}$ shall be upper bounded by the bandwidth b_e, for all $e \in E$ and all $i = 0, 1, ..., T - 1$, where $T = D/R$.

Lemma 5.3. *Given any instance of MPA (or MAA, MMD), suppose R is an arbitrary throughput satisfying $R \in [R_l, R_u]$ and $D/R \in \mathbb{Z}^+$. Let us assume M to be an arbitrary integer. Then the problem of whether there exists a feasible periodically repeated solution f with $\mathcal{T}(f) = D/R$ and $\mathcal{M}(f) \le M$ is feasible if and only if the value of the optimal solution to the linear program* (13) *is no smaller than D.*

Proof. Refer to Appendix 9.7 of our technical report [12] □

To obtain $\mathcal{M}^R$, Lem. 5.3 suggests that we can use binary search to obtain the minimal integer $M^* \in [0, M_U]$, under which the linear program (13) outputs a feasible flow with a value no smaller than D, and it is clear that the achieved M^* shall be the $\mathcal{M}^R$ (see Algorithm 2). Note that to construct the expanded network, we need a $M_U \ge \mathcal{M}^R$. We remark that M_U must exist, e.g., we can set $M_U = |V| \cdot (d_{\max} + D/R_l)$ with $d_{\max} = \max_{e \in E} d_e$, since for any path $p \in P$ and any offset $\vec{u} \in \vec{\mathcal{U}}$ that corresponds to p, the following holds for any periodically repeated solution: (i) $|V| \cdot d_{\max}$ is an upper bound of the aggregate delay experienced by passing all the links that belong to p, since p is simple, and (ii) $|V|D/R_l$ is an upper bound of the aggregate data-holding delay at all the nodes that belong to p, due to our Lem. 3.1.

Algorithm 2 Obtain $\mathcal{M}^R$ and the corresponding solution

input: $G = (V, E)$, R, D, M_U, s, r
output: f, $\mathcal{M}$
procedure
  $f = f_t = \text{NULL}$, $\mathcal{M} = +\infty$, $LB = 0$, $UB = M_U$
  Obtain G_{exp} by Algorithm 1 with (G, M_U)
  while $LB \le UB$ **do**
    $M = \lceil (LB + UB)/2 \rceil$
    f_t is the solution by solving the linear program (13) with input $(G_{\text{exp}}, R, D, M, s, r)$
    if the objective of f_t is no smaller than D **then**
      $f = f_t$, $\mathcal{M} = M$, $UB = M - 1$
    else
      $LB = M + 1$
  return f, $\mathcal{M}$

Lemma 5.4. *Suppose $\mathcal{L}$ is the input size of the instance of linear program* (13)*, then the time complexity of Algorithm 2 is $O(|E|^3 M_U^3 \mathcal{L} \log M_U)$.*

Proof. Refer to Appendix 9.8 of our technical report [12] □

Lem. 5.4 shows that our Algorithm 2 has a pseudo-polynomial time complexity, because of the pseudo-polynomial size of the expanded network: (i) considering $M_U \le |V| \cdot (d_{\max} + D/R_l)$, the time complexity is polynomial with the numeric value of $d_{\max}$ and D/R_l, but (ii) it is exponential with the bit length of $d_{\max}$ and D/R_l.

5.4 Use Algorithm 2 to Solve MPA and MAA Optimally in a Pseudo-Polynomial Time

We remark that it is challenging to obtain the optimal throughput $R_p \in [R_l, R_u]$ (resp. $R_a \in [R_l, R_u]$) that minimizes peak AoI (resp. average AoI), due to the following observation.

Lemma 5.5. *Both Λ_p^R and Λ_a^R are non-monotonic, non-convex, and non-concave with R theoretically.*

Proof. Refer to Appendix 9.9 of our technical report [12] □

Thus to solve MPA (resp. MAA) optimally, we need to enumerate Λ_p^R (resp. Λ_a^R) for all $R \in [R_l, R_u], D/R \in \mathbb{Z}^+$, and obtain the optimal one that achieves minimal peak AoI (resp. minimal average AoI). It is clear that we can use Algorithm 2 to achieve Λ_p^R and Λ_a^R. Therefore, we suggest to solve MPA (resp. MAA) optimally using Algorithm 2 by enumerating all possible throughputs.

We remark that our proposed enumerating approach has a pseudo-polynomial time complexity. As shown in Lem. 5.4, Algorithm 2 has a pseudo-polynomial time complexity to obtain Λ_p^R and Λ_a^R. Now considering that the number of the enumerated throughputs is $D/R_l - D/R_u + 1$ which is pseudo-polynomial with D/R_l, using Algorithm 2 to solve MPA and MAA optimally by enumeration has a pseudo-polynomial time complexity, too.

Overall, we have the following theorem for MPA and MAA.

Theorem 5.6. *MPA and MAA are NP-hard in the weak sense.*

Proof. It is a direct result from Lem. 5.2 and our proposed optimal algorithm which has a pseudo-polynomial time complexity. □

6 AN APPROXIMATION FRAMEWORK

As discussed in Sec. 4, the peak/average AoI of the solution minimizing maximum delay is within a bounded gap as compared to optimal. Thus it is natural to use approximate solutions to the problem of minimizing maximum delay as approximate solutions to our problems of minimizing AoI. However, this idea is non-trivial, considering that as discussed in Sec. 2, existing maximum delay minimization problems (i.e., the quickest flow problem and the min-max-delay flow problem) are just special cases of the maximum-delay-minimization counterpart of our AoI minimization problems. This is because they assume a fixed task activation period, which is quite different from our problems that assume the task activation period to be decision variables. In this section, we overcome the challenge, and propose a framework that can adapt any polynomial-time approximation algorithm of the min-max-delay flow problem to solve our MPA and MAA approximately in a polynomial time.

For any feasible periodically repeated solution f to MPA and MAA achieving a throughput of R, it should send D amount of data from s to G every D/R slots, meeting link bandwidth constraints. According to the definition of the min-max-delay flow problem (refer to [13]), for any feasible solution $\mathbf{f}$ to the min-max-delay flow problem achieving a throughput of R, it should send R amount of data from s to G at each slot, meeting link bandwidth constraints. Because it is clear that this $\mathbf{f}$ can send D amount of data from s to G every D/R slots, meeting link bandwidth constraints, we observe that $\mathbf{f}$ is a special case of f.

Let us denote a feasible instance of MPA (resp. MAA) characterized by (G, s, r, R_l, R_u, D) as $\mathbf{MPA}(R_l, R_u, D)$ (resp. $\mathbf{MAA}(R_l, R_u, D)$). And denote the corresponding min-max-delay flow problem instance, which is defined by the same G, s, r, but with a throughput requirement of R, as $\mathbf{MMD1}(R)$ (note as discussed in Sec. 2, min-max-delay flow problem assumes a fixed task activation period of 1, and thus a fixed throughput requirement, but MPA and MAA assume both a minimum and maximum throughput requirement). We have the following lemma.

Lemma 6.1. *Given any MPA(R_l, R_u, D) (resp. MAA(R_l, R_u, D)), suppose $R \in [R_l, R_u], D/R \in \mathbb{Z}^+$ is an arbitrary feasible throughput for it. Then MMD1(R) must be feasible. Moreover, suppose $\boldsymbol{f}(R)$ is an arbitrary feasible solution to MMD1(R), it holds that $\boldsymbol{f}(R)$ must be a feasible periodically repeated solution to MPA(R_l, R_u, D) (resp. MAA(R_l, R_u, D)) with the following*

$$\mathcal{M}(\boldsymbol{f}(R)) \;=\; \hat{\mathcal{M}}(\boldsymbol{f}(R)) + D/R - 1,$$

where $\hat{\mathcal{M}}(\boldsymbol{f})$ is the maximum delay of $\boldsymbol{f}$ with MMD1(R).

Proof. Refer to Appendix 9.10 of our technical report [12] □

Lem. 6.1 suggests that any feasible solution to the min-max-delay flow problem achieving a throughput of R is a feasible periodically repeated solution to the corresponding MPA and MAA also achieving a throughput of R. But we remark that even for the optimal solution to the min-max-delay flow problem, its peak AoI (resp. average AoI) can be strictly greater than the minimal peak AoI (resp. minimal average AoI) with a throughput of R, i.e., than Λ_p^R (resp. Λ_a^R). This is because when we look at a solution to the min-max-delay flow problem from the perspective of MPA and MAA, it always sends R amount of data from s to G at each slot, which is a special case of feasible solutions to MPA and MAA. In fact, MPA and MAA allow various amount of data to be sent to G at each slot, as long as a total of D amount of data can be sent every D/R slots.

Lem. 6.1 suggests that we can use the solution to the min-max-delay flow problem as a solution to our MPA (resp. MAA). As it is easy to prove that if MMD1(R_1) is feasible, MMD1(R_2) must be feasible given any $0 < R_2 \le R_1$ (see the proof to the following theorem), a direct result of Lem. 6.1 is that R_l must be a feasible throughput for MPA(R_l, R_u, D) (resp. MAA(R_l, R_u, D)). Therefore, it is clear that solving MMD1(R_l) must output a feasible solution to MPA(R_l, R_u, D) (resp. MAA(R_l, R_u, D)). In the following theorem, we further prove that any approximate solution to MMD1(R_l) must be an approximate solution to MPA(R_l, R_u, D) (resp. MAA(R_l, R_u, D)), with bounded optimality loss. For easier reference, we denote an arbitrary α-approximation algorithm of the min-max-delay flow problem as **ALG-MMD1**(α).

Theorem 6.2. *Given any MPA(R_l, R_u, D) and MAA(R_l, R_u, D) where $D/R_l \in \mathbb{Z}^+$, $D/R_u \in \mathbb{Z}^+$, suppose we use ALG-MMD1(α) to solve the corresponding MMD1(R_l). Then it must give an α-approximate solution $\boldsymbol{f}_\alpha(R_l)$ to MMD1(R_l). Moreover, $\boldsymbol{f}_\alpha(R_l)$ must be a feasible periodically repeated solution to MPA(R_l, R_u, D) and MAA(R_l, R_u, D), with an approximation ratio of $(\alpha + \mathrm{c})$ where c is defined below*

$$\mathrm{c} = \begin{cases} 2 \cdot R_u/R_l, & \text{for MPA}(R_l, R_u, D), \\ 3 \cdot R_u/R_l, & \text{for MAA}(R_l, R_u, D). \end{cases} \tag{14}$$

Proof. Refer to Appendix 9.11 of our technical report [12] □

Thm. 6.2 shows that for any α-approximation algorithm of the min-max-delay flow problem, we can directly use it to solve MPA and MAA approximately instead, with approximation ratios determined by α, R_l, and R_u. Note that approximation algorithms for

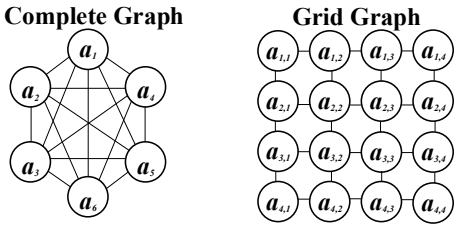
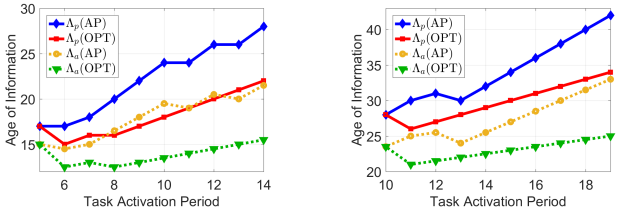


Figure 3: Two simulated network topologies.



(a) Complete graph with $D = D_1$. (b) Grid graph with $D = D_2$.

Figure 4: Simulation results of two representative instances on the typical graphs.

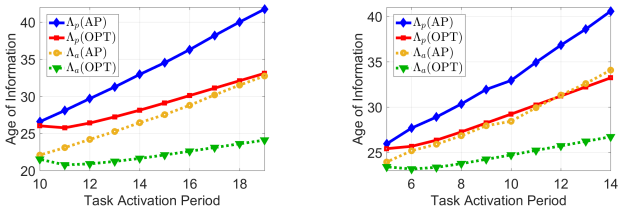
the min-max-delay flow problem exist in the literature, e.g., Misra *et al.* [13] have designed a $(1+\epsilon)$-approximation algorithm, where ϵ can be an arbitrary user-defined positive number.

7 PERFORMANCE EVALUATION

We evaluate the empirical performance of our proposed approaches, by simulating (i) two typical network topologies shown in Fig. 3, and (ii) nine random network topologies generated by well-known random graph generation models. All networks are modeled as undirected graphs, where each undirected link is treated as two directed links that operate independently. Each link delay is randomly generated from $\{1, 2, 3, 4, 5\}$, and each link bandwidth is randomly generated from $\{10, 20, 30, 40, 50\}$. Given a network, we consider two different D with $D_1 = 5 \cdot \mathcal{D}$ and $D_2 = 10 \cdot \mathcal{D}$, where $\mathcal{D}$ is the maximum amount of data that can be streamed from sender to receiver with a unit task activation period. Note that this $\mathcal{D}$ is also the maximum throughput that can be achieved in each simulation, based on Lem. 6.1. Thus 5 (resp. 10) is the minimal possible task activation period for simulations with $D = D_1$ (resp. with $D = D_2$). In each simulation, we consider ten different task activation periods (thus ten different throughputs), where $\mathcal{T}(f) \in \{5, 6, ..., 14\}$ (resp. $\mathcal{T}(f) \in \{10, 11, ..., 19\}$) for simulations with $D = D_1$ (resp. with $D = D_2$). The ALG-MMD1(α) used by our approximation framework is the $(1+\epsilon)$-approximation algorithm from [13] and we set $\epsilon = 1$. Our test environment is an Intel Core i5 (2.40 GHz) processor with 8 GB memory. All the experiments are implemented in C++ and linear programs are solved using *CPLEX* [5].

7.1 Simulations on Typical Networks

The two typical network topologies simulated are (i) a complete graph with 6 nodes and 15 undirected links, and (ii) a grid graph with 16 nodes and 24 undirected links. The complete graph topology represents a fully-connected and thus ideal network structure, while



(a) Complete graph with $D = D_2$. (b) Grid graph with $D = D_1$.

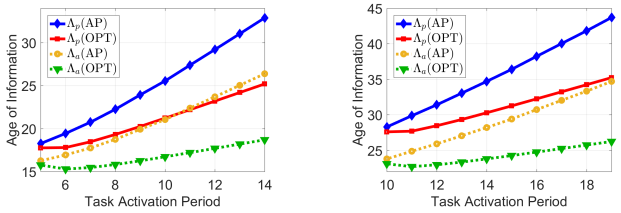
Figure 5: Simulation results in average of 200 instances on the typical graphs.

the grid graph topology represents a distributed network structure. In Fig. 3, for the complete network, we assume the sender to be a_1 and the receiver to be a_6, and for the grid network, we assume the sender to be $a_{1,1}$ and the receiver to be $a_{4,4}$.

First, we give the AoI results of one representative instance simulated on the complete graph with $D = D_1$ (resp. on the grid graph with $D = D_2$) in Fig. 4(a) (resp. Fig. 4(b)), where for each throughput R (thus for each task activation period D/R), Λ_p(AP) (resp. Λ_a(AP)) is the peak AoI (resp. average AoI) of the solution of ALG-MMD1(2) with a throughput requirement of R, while Λ_p(OPT) (resp. Λ_a(OPT)) is exactly Λ_p^R (resp. Λ_a^R), which is the peak AoI (resp. average AoI) of the solution of our Algorithm 2.

From Fig. 4(a), empirically we verify (i) Lem. 5.1, where the task activation period of 6 (thus the throughput of $D_1/6$) achieving the optimal peak AoI is different from that of 8 (resp. that of $D_1/8$) achieving the optimal average AoI, and (ii) Lem. 5.5, where the minimal peak AoI (resp. minimal average AoI) is non-monotonic, non-convex, and non-concave with throughput.

Considering that we generate link bandwidths and delays randomly, next, we simulate 100 instances of the complete network with $D = D_2$ (resp. 100 instances of the grid network with $D = D_1$), and present the AoI results in average in Fig. 5(a) (resp. Fig. 5(b)). (i) Empirically, we observe that Λ_p^R and Λ_a^R are "almost" increasing with throughput R. Note that for an instance of MPA (resp. MAA), the peak AoI (resp. average AoI) of our approximation framework is the Λ_p(AP) (resp. Λ_a(AP)) corresponding to the smallest throughput (thus the largest task activation period), while the peak AoI (resp. average AoI) of our optimal algorithm is the smallest peak AoI (resp. average AoI) among those achieved by all possible throughputs (thus all possible task activation periods). (ii) Empirically, we observe that our optimal algorithm obtains a 3.8% peak AoI reduction (resp. 3.2% average AoI reduction) as compared to our approximation framework, when the number of possible throughputs (thus the range of task activation period) of an instance of MPA (resp. MAA) increases by 1. However, (iii) given a specific throughput, the average running time of ALG-MMD1(2) (resp. of Algorithm 2) is 0.06s (resp. 0.10s). Therefore for an instance of MPA and MAA, the running time of our approximation framework is a constant 0.06s (directly run ALG-MMD1(2) with the smallest throughput requirement), while that of our optimal algorithm increases by 0.10s when the number of possible throughputs increases by 1 (enumerate AoIs achieved by all possible throughputs to figure out the optimal).



(a) Instances with $D = D_1$. (b) Instances with $D = D_2$.

Figure 6: Simulation results in average of random graphs that are generated by *SNAP* [9].

7.2 Simulations on Random Networks

We also use *SNAP* [9] to randomly generate nine network topologies, where three of them follow Erdos-Renyi model, another three of them follow Watts-Strogatz model, and the remaining three of them follow Copying model. For the model-related parameters, we set $n = 20$ which is the number of nodes and $m = 50$ which is the number of (undirected) links. Besides, we use default values both of the degree parameter $k = 3$ and of the degree-exponent parameter $p = 0.1$. Definitions of those graph generation models and associated parameters are given by [9].

For each of the nine topologies, we run 100 simulation instances respectively with $D = D_1$ and with $D = D_2$. Note that for each simulation instance, the sender and the receiver are randomly selected. The simulated AoI results on random networks (Fig. 6) is very similar to that on typical networks (Fig. 5). When the range of task activation period of an instance increases by 1, (i) our optimal algorithm obtains a 4.3% peak AoI reduction (resp. 4.0% average AoI reduction) as compared to our approximation framework; (ii) our approximation framework has a constant running time of 0.06s, while the running time of our optimal algorithm increases by 0.11s.

8 CONCLUSION

We consider a scenario where a sender periodically sends a batch of data to a receiver over a multi-hop network using multiple paths. We study problems of minimizing peak/average AoI, by jointly optimizing (i) the throughput subject to throughput requirements, and (ii) the multi-path routing strategy. The consideration of batch generation and multi-path communication differentiates our study from existing ones. First we show that our problems are NP-hard but only in the weak sense, as we develop a pseudo-polynomial-time optimal algorithm. Next, we show that minimizing AoI is "largely" equivalent to minimizing maximum delay, as any optimal solution of the latter is an approximate solution to the former, with bounded optimality loss. We leverage this understanding to design a framework to adapt any polynomial-time α-approximation algorithm of the maximum delay minimization problem to solve our AoI minimization problems, with an approximation ratio of $\alpha + c$. The framework suggests a new avenue for developing approximation algorithms for minimizing AoI in multi-path communications. We conduct extensive simulations over various network topologies to empirically validate the effectiveness of our approach.

REFERENCES

[1] Shi Bai, Weiyi Zhang, Guoliang Xue, Jian Tang, and Chonggang Wang. 2012. DEAR: Delay-bounded energy-constrained adaptive routing in wireless sensor networks. In *Proc. IEEE Int'l Conf. Computer Communications.*

[2] Lei Deng, Chih-Chun Wang, Minghua Chen, and Shizhen Zhao. 2017. Timely wireless flows with general traffic patterns: Capacity region and scheduling algorithms. *IEEE/ACM Trans. Networking* 25, 6 (2017), 3473–3486.

[3] I-H Hou, Vivek Borkar, and PR Kumar. 2009. A theory of QoS for wireless. In *Proc. IEEE Int'l Conf. Computer Communications.*

[4] Longbo Huang and Eytan Modiano. 2015. Optimizing age-of-information in a multi-class queueing system. In *Proc. IEEE Int'l Sym. Information Theory.*

[5] IBM. 2017. Cplex Optimizer. (2017). Available at https://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/.

[6] Igor Kadota, Abhishek Sinha, and Eytan Modiano. 2018. Optimizing age of information in wireless networks with throughput constraints. In *Proc. IEEE Int'l Conf. Computer Communications.*

[7] Sanjit Kaul, Marco Gruteser, Vinuth Rai, and John Kenney. 2011. Minimizing age of information in vehicular networks. In *Proc. IEEE Communications Society Conf. Sensor, Mesh and Ad Hoc Communications and Networks.*

[8] Sanjit Kaul, Roy Yates, and Marco Gruteser. 2012. Real-time status: How often should one update?. In *Proc. IEEE Int'l Conf. Computer Communications.*

[9] Jure Leskovec and Rok Sosič. 2016. SNAP: A General-Purpose Network Analysis and Graph-Mining Library. *ACM Trans. Intelligent Systems and Technology* 8, 1 (2016), 1.

[10] Maokai Lin and Patrick Jaillet. 2015. On the quickest flow problem in dynamic networks: a parametric min-cost flow approach. In *Proc. ACM-SIAM Sym. Discrete algorithms.*

[11] Qingyu Liu, Lei Deng, Haibo Zeng, and Minghua Chen. 2018. A Tale of Two Metrics in Network Delay Optimization. In *Proc. IEEE Int'l Conf. Computer Communications.*

[12] Qingyu Liu, Haibo Zeng, and Minghua Chen. 2018. Minimizing Age-of-Information with Throughput Requirements in Multi-Path Network Communication. *arXiv preprint arXiv:1811.12605* (2018). Available at https://arxiv.org/abs/1811.12605.

[13] Satyajayant Misra, Guoliang Xue, and Dejun Yang. 2009. Polynomial time approximations for multi-path routing with bandwidth and delay constraints. In *Proc. IEEE Int'l Conf. Computer Communications.*

[14] Xukan Ran, Haoliang Chen, Zhenming Liu, and Jiasi Chen. 2017. Delivering deep learning to mobile devices via offloading. In *ACM Workshop Virtual Reality and Augmented Reality Network.*

[15] Masahide Saho and Maiko Shigeno. 2017. Cancel-and-tighten algorithm for quickest flow problems. *Networks* 69, 2 (2017), 179–188.

[16] Yin Sun, Elif Uysal-Biyikoglu, and Sastry Kompella. 2018. Age-optimal updates of multiple information flows. *arXiv preprint arXiv:1801.02394* (2018).

[17] Yin Sun, Elif Uysal-Biyikoglu, Roy D Yates, C Emre Koksal, and Ness B Shroff. 2017. Update or wait: How to keep your data fresh. *IEEE Trans. Information Theory* 63, 11 (2017), 7492–7508.

[18] Rajat Talak, Igor Kadota, Sertac Karaman, and Eytan Modiano. 2018. Scheduling policies for age minimization in wireless networks with unknown channel state. In *Proc. IEEE Int'l Sym. Information Theory.*

[19] Rajat Talak, Sertac Karaman, and Eytan Modiano. 2017. Minimizing age-of-information in multi-hop wireless networks. In *Proc. IEEE Allerton Conf. Communication, Control, and Computing.*

[20] Rajat Talak, Sertac Karaman, and Eytan Modiano. 2018. Can Determinacy Minimize Age of Information? *arXiv preprint arXiv:1810.04371* (2018).

[21] Rajat Talak, Sertac Karaman, and Eytan Modiano. 2018. Optimizing information freshness in wireless networks under general interference constraints. In *Proc. ACM Int'l Sym. Mobile Ad Hoc Networking and Computing.*

[22] Chih-Chun Wang and Minghua Chen. 2017. Sending perishable information: Coding improves delay-constrained throughput even for single unicast. *IEEE Trans. Information Theory* 63, 1 (2017), 252–279.

[23] Jiantao Wang, Lun Li, Steven H Low, and John C Doyle. 2005. Cross-layer optimization in TCP/IP networks. *IEEE/ACM Trans. Networking* 13, 3 (2005), 582–595.

[24] Meng Wang, Chee Wei Tan, Weiyu Xu, and Ao Tang. 2011. Cost of not splitting in routing: Characterization and estimation. *IEEE/ACM Trans. Networking* 19, 6 (2011), 1849–1859.