

Minimizing AoI with Throughput Requirements in Multi-Path Network Communication

Qingyu Liu, Haibo Zeng, Minghua Chen

Abstract—We consider a single-unicast networking scenario where a sender periodically sends a batch of data to a receiver over a multi-hop network, possibly using multiple paths. We study problems of minimizing peak/average Age-of-Information (AoI) subject to throughput requirements based on a stylized deterministic model in this scenario. 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 [2] 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. Furthermore, we show that our results can be extended to the multiple-unicast setting. Simulations over various network topologies validate the effectiveness of our approach. Our results make a major advance to optimizing AoI in multi-path communication, and hence can be of broad interest to the networking research community.

Index Terms—Age-of-information, AoI minimization, multi-path communication, time-critical network flow

I. INTRODUCTION

Age-of-Information (AoI) is a critical networking performance metric for periodic services that require timely transmissions. Kaul *et al.* [3] measures the AoI as the time that elapsed since the last received update was generated. In this paper, we study AoI-minimization problems of supporting a periodic transmission task under a single-unicast networking scenario. Specifically, the task requires a sender to send a

batch of data (packets) periodically to a receiver over a multi-hop network, possibly using multiple paths. We use a stylized deterministic model to model the network. 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.

Our study is motivated by leveraging a network platform with limited resources to support periodic transmission tasks that are sensitive to both throughputs and end-to-end delays. Below we provide a specific example: offloading real-time image-processing tasks in edge computing.

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 and frequently. The widely-adopted solution is to leverage nearby edge servers for workload offloading. For example, Ran *et al.* [24] develop an Android application of real-time object detection. If running locally on the phone for 30 minutes, it processes images at the rate of 5 FPS 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 [24] we note that the majority (over 95%) of the total delay of running tasks remotely is the networking delay. Therefore, to offload image-processing tasks to an edge computing platform for processing, 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 for real-time actions, e.g., at the end users or the edge computing nodes, it is important to minimize AoI.

Our Contributions. In this paper, we consider the scenario that a sender is required to periodically send a batch of data to a receiver over a multi-hop network, possibly using multiple paths. As an example, in edge computing a real-time image-processing task may require a mobile device to transmit an image periodically to an edge server first over a one-hop wireless network and then over a multi-hop wired network. We study AoI-minimization problems subject to throughput requirements based on a stylized deterministic network model. Table I summarizes the comparison of our AoI study with existing ones, and the details are provided in Section II. The consideration of batch generation and multi-path communication differentiates our AoI study from existing ones. In this

Manuscript received October 24, 2020; revised July 17, 2021, October 25, 2021, and December 3, 2021; accepted December 6, 2021; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor Krishna Jagannathan. The work of Qingyu Liu and Haibo Zeng was supported in part by NSF Grant No. 479987. The work of Minghua Chen was supported in part by a Start-up Grant from School of Data Science (Project No. 9380118), City University of Hong Kong, and a General Research Fund from Research Grants Council, Hong Kong (Project No. 14207520). A preliminary version of the work was presented at the ACM International Symposium on Mobile Ad Hoc Networking and Computing 2019 [1]. (Corresponding authors: Qingyu Liu; Haibo Zeng; Minghua Chen.)

Qingyu Liu and Haibo Zeng are with the Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24060 USA (e-mail: qyliu14@vt.edu; hbzeng@vt.edu).

Minghua Chen is with the School of Data Science, City University of Hong Kong, Hong Kong, China (e-mail: minghua.chen@cityu.edu.hk).

This article has supplementary downloadable material available at <http://ieeexplore.ieee.org>, provided by the authors.

Digital Object Identifier

TABLE I
COMPARE OUR Aol STUDY WITH EXISTING ONES.

		[3]–[10]	[11]	[12]	[13], [14]	[15]–[17]	[18]–[20]	[21]	[22]	[23]	Our work
Objective of Optimization	Minimize peak Aol	X	✓	✓	X	X	✓	✓	✓	X	✓
	Minimize average Aol	✓	X	✓	✓	✓	✓	✓	✓	✓	✓
Design Space of Optimization	Multi-path routing strategy	X	X	X	X	X	X	X	X	X	✓
	Information generation rate*	✓	✓	✓	✓	X	X	✓	✓	X	✓
	Link scheduling policy	X	X	X	✓	✓	✓	X	✓	✓	X
	Queuing disciplines	X	✓	✓	X	X	X	X	X	✓	X

Note. *: Under our system model, including the information generation rate in the design space is equivalent to including the throughput.

paper, we claim the following specific contributions.

▷ Comparing minimizing peak/average Aol with minimizing maximum delay: (i) we show that the throughput of the Aol-minimal solution can be different from, but is always no smaller than, that of the maximum-delay-minimal solution (Lemma 2). This result is consistent with our observation that Aol is a metric simultaneously considering maximum delay and throughput; (ii) we show that the Aol of the maximum-delay-minimal solution can be larger than the minimal Aol (Lemma 2), but the gap is bounded (Lemma 3).

▷ Comparing minimizing peak Aol with minimizing average Aol: (i) we prove that the average Aol of the peak-Aol-minimal solution can be larger than the minimal average Aol, and vice versa, but both gaps are bounded (Lemma 4); (ii) we show that the throughput (resp. maximum delay) of the peak-Aol-minimal solution can be different from, but is always no smaller than, that of the average-Aol-minimal solution (Lemma 4). Thus, as compared to minimizing average Aol, minimizing peak Aol may carry more flavor on throughput and less on maximum delay.

▷ We observe that both minimizing peak Aol and minimizing average Aol are challenging, because (i) we prove that theoretically both minimal peak Aol and minimal average Aol are non-monotonic, non-convex, and non-concave with throughput (Lemma 8), and (ii) we prove that both problems are NP-hard (Lemma 5), but in the weak sense (Theorem 1), as we design an algorithm to solve them optimally in a pseudo-polynomial time (Section V-D).

▷ We leverage our understanding on comparing Aol with maximum delay to develop an approximation framework (Theorem 2). It can build upon any α -approximation algorithm of the maximum delay minimization problem, e.g., the algorithm in [2] with $\alpha = 1 + \epsilon$ given any user-defined $\epsilon > 0$, to construct an $(\alpha + c)$ -approximate solution for minimizing Aol. Here c is a constant depending on the throughput requirements.

▷ We further show that our results can be extended to the multiple-unicast setting. Specifically, our framework can also use any existing approximation algorithm of the maximum delay minimization problem to construct an approximate solution for minimizing Aol in the multiple-unicast setting.

▷ To the best of our knowledge, our work presents the first step to minimize Aol by optimizing multi-path routing. Therefore, although we use a stylized deterministic model to model the network, our results make a major advance to optimizing Aol in multi-path communication. Hence, our results can be of broad interest to the networking community.

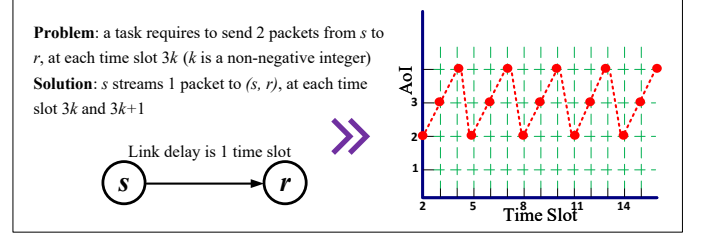


Fig. 1. An illustrative example of our batch-based Aol. Sender s generates a batch of two packets at each slot $3 \cdot k$ with k to be a non-negative integer. It sends the two packets one-by-one over link (s, r) to the receiver r ; the link transmission incurs one-slot delay. Our batch-based Aol drops 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 $3k + 2$, the batch-based Aol at each slot $3k + 2$ drops to 2, i.e., the elapsed time since the generation of the last received batch. The batch-based Aol at all the other slot grows linearly.

II. RELATED WORK

Since its introduction in [3], Aol has attracted active research efforts in the research community (see [25] for a comprehensive review). One main line of research is to minimize Aol under certain resource constraints, where the resource ranges from bandwidth (e.g., [13]–[20], [22], [23]), to energy (e.g. [6]–[10]), and to throughput (e.g., [3]–[5], [11]–[14], [22]). We compare these Aol studies with our work in Table I. In summary, we differ from them in two aspects, i.e., the problem design space and the Aol definition.

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., [26], [27], have shown that multi-path routing can provide better QoS, e.g., larger throughput, than the single-path routing. To our best knowledge, we are the first to optimize the multi-path routing strategy to minimize Aol.

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

It is worth highlighting that Kadota *et al.* [13], [14] minimize Aol with throughput requirements in the single-hop network. In this paper we minimize Aol with throughput

requirements in the multi-hop network. However, our results are not extensions of the results in [13], [14], as the problem studied in [13], [14] is fundamentally different from ours: Kadota *et al.* [13], [14] consider a single-hop network with stochastic parameters involved (transmitting data over the link only succeed with certain probability). As multiple senders share the same link to transmit data to a common receiver, Kadota *et al.* [13], [14] minimize AoI by optimizing link scheduling policy that determines which sender will use the link at each time. In contrast, there is no randomness in the study presented in this paper and there is no need to consider link scheduling across senders. In this paper, we use a stylized deterministic model to model the multi-hop network. We minimize AoI by optimizing multi-path routing.

It is also worth highlighting that there exist some AoI studies considering multi-hop networks. For example, Talak *et al.* [19] develop link scheduling policies for minimizing AoI under wireless link interference constraints in a multi-hop network. However, they only consider single-hop information flows in a multi-hop network. The extension to multi-hop information flows is studied in [22]. Ayan *et al.* [28] derive closed form expressions for AoI in a multi-hop network. Farazi *et al.* [21] derive lower bounds for the minimal AoI, and design schedules for sources to generate information to minimize AoI in a multi-hop network. Our work is fundamentally different from studies in [19], [21], [22], [28], because we minimize AoI by optimizing multi-path routing, but they minimize AoI by optimizing link scheduling policy or information generation rate (as illustrated in Table I).

In addition to minimizing AoI under resource constraints, another line of AoI research has been focused on modeling, analysis, and optimization of AoI (e.g., [29], [30]). There are also some other branches on AoI research, such as scheduling under AoI constraints (e.g., [31], [32]), game theory for AoI (e.g., [33], [34]), and channel coding for AoI (e.g., [35], [36]). Those AoI studies are less relevant to our work.

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 amount of data in the batch divided by throughput) is a decision variable. In contrast, to our best knowledge, existing 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 [37], [38]. 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. It assumes that the task activation period is infinitely large. It is polynomial-time solvable under our setting [37].

Min-max-delay flow problem [2], [39], [40]. 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 assumes that the task activation period is one unit of time. It is NP-hard under our setting [2].

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 $R \rightarrow 0$ 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 Lemma 1 introduced later, given $R_l = R_u$, minimizing AoI is equivalent to minimizing maximum delay. This implies that the quickest flow problem ($R_l = R_u \rightarrow 0$) 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 [37] and approximation algorithms for the min-max-delay flow problem [2] 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 Lemma 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.¹

In the literature there exist some other time-aware communication studies. One line is the throughput-optimal routing policy design with delay taken into account: The celebrated Back Pressure (BP) algorithm [41] is a routing policy to achieve maximum throughput. But its delay can be large, particularly for lightly loaded networks where there is not enough “pressure” to push data. Subsequently, many refinements have been proposed to improve its delay performance. For example, Sinha and Modiano [42] develop the Universal Max-Weight policy for achieving maximum throughput with reduced delay. Although there exist throughput-optimal policies that are delay-aware, they do not explicitly consider delay as a minimization objective or a constraint ([42] can improve delay performance because it does not allow packet cycling). Therefore, they are not directly related to our study.

Other time-aware studies in the literature include [43]–[45]. Hou *et al.* [43] propose scheduling policies for a set of senders to be feasible with respect to the delay constraint, throughput constraint, and wireless channel reliability constraint. In [44], Hou *et al.* extend their previous work and study the utility maximization problem. Deng *et al.* [45] further conduct a complete study on the similar timely wireless flow problem but assuming a more general traffic pattern. Those studies [43]–[45] are of little relevance to our problems, because their focus is the wireless link scheduling policy optimization. Differently, we focus on optimizing the multi-path routing strategy.

III. SYSTEM MODEL

A. Preliminary

We consider a multi-hop network modeled as a directed graph $G \triangleq (V, E)$ with $|V|$ nodes and $|E|$ links. We assume

¹Note that although our objective is to minimize AoI, in this paper we also solve the problem of minimizing maximum delay which has not been studied in the literature. This is because minimizing maximum delay is a special case of minimizing AoI in our setting (based on Lemma 1, minimizing maximum delay is equivalent to minimizing AoI if $R_l = R_u$).

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 $k \cdot T$ for each $k \in \mathbb{Z}, k \geq 0$ (k is a non-negative integer), and is required to transmit them to a receiver node $r \in V \setminus \{s\}$, possibly using multiple paths. Because we assume no data loss during transmission, clearly the **throughput** incurred by a task activation period of T is D/T .

We aim to obtain a multi-path routing solution that is *periodically repeated* to periodically send the batch of data. 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 $|p|$. Suppose $p = \langle v_1, v_2, \dots, v_{|p|} \rangle$ where $\{v_i \in V, i = 1, 2, \dots, |p|\}$ are the nodes on p ($v_1 = s$ and $v_{|p|} = r$) and $\{e_{i-1} = (v_{i-1}, v_i) \in E, i = 2, 3, \dots, |p|\}$ are the links belonging to p . Then we can use $x^p(\vec{u})$ with $\vec{u} = \langle u_0, u_1, u_2, \dots, u_{|p|} \rangle$ to represent that we 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 $k \cdot T$ onto link (v_i, v_{i+1}) at time $k \cdot T + u_i$. By the definition of $x^p(\vec{u})$, it is clear that $u_i - (u_{i-1} + d_{e_{i-1}})$ is the data-holding delay at the node v_i (we assume $u_0 = 0$ and $d_{e_0} = 0$). Since the data-holding delay must be non-negative, the following holds for each $i = 1, 2, \dots, |p|$:

$$u_i \in \mathbb{Z} \text{ and } u_i - u_{i-1} \geq d_{e_{i-1}}.$$

Here we also highlight that the data-holding delay shall be upper bounded by $T - 1$, i.e., $u_i - u_{i-1} \leq d_{e_{i-1}} + T - 1$ (proved in Part A of supplementary materials). We denote $\vec{U}$ as the set of all $\vec{u}$'s. With the definition of $x^p(\vec{u})$, we define a periodically repeated solution f as the assigned amount of data over P at the time offset $\vec{U}$:²

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

We denote $\mathcal{T}(f)$ as the task activation period of f .

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 $e_i \in p$, 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 $\mathcal{A}^p(\vec{u})$ ³,

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

²Note that for a multi-path routing solution f , we do not have node-disjoint or link-disjoint requirements on different paths $p_1 \in P$ and $p_2 \in P$ that carry traffic (i.e., with $x^{p_1}(\vec{u}) > 0$ and $x^{p_2}(\vec{u}) > 0$).

³The delay of a path in our system depends on the link delay of each link in the path, as well as on the data-holding delay of each node in the path. These delay definitions have been used in many related works, e.g., [2], [37].

One important time-aware 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.,

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

Another important time-aware metric is the **Aol**, denoted as $\mathcal{I}(f, t)$. Aol measures the time that elapsed since the generation of the task period that was most recently delivered to the receiver. Specifically, we define the Aol at time t as

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

where $\pi_t(f)$ is the generation time of the task period that was most recently delivered to r by time t , or equivalently the starting time of the most recent task period with all its D amount of data arriving at r no later than t , i.e.,

$$\pi_t(f) \triangleq \max_{k \in \mathbb{Z}, k \geq 0} \{k \cdot \mathcal{T}(f) : k \cdot \mathcal{T}(f) + \mathcal{M}(f) \leq t\}.$$

For simplicity, we assume that $\mathcal{I}(f, 0) = 1$.

B. Problem Definition

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

$$\Lambda_p(f) \triangleq \limsup_{\tau \rightarrow \infty} \{\mathcal{I}(f, t) : t = 0, 1, \dots, \tau - 1\},$$

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

$$\Lambda_a(f) \triangleq \lim_{\tau \rightarrow \infty} \frac{1}{\tau} \cdot \sum_{t=0}^{\tau-1} \mathcal{I}(f, t).$$

Our problems of finding a periodically repeated solution f to minimize Aol 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 $R_u \in \mathbb{R}$):

$$\sum_{\forall p \in P} \sum_{\forall \vec{u} \in \vec{U}} x^p(\vec{u}) = D, R_l \leq D/\mathcal{T}(f) \leq R_u, \text{ and } \mathcal{T}(f) \in \mathbb{Z}^+. \quad (1)$$

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

It is understandable to optimize Aol under a minimum throughput requirement R_l . The reason why we also consider a maximum throughput requirement R_u is as follows: As we assume no data loss during transmission, the throughput is D/T . Clearly this D/T is also the rate of generating data at the sender (i.e., sender's sampling rate). Therefore, in our problem R_u is equivalent to a constraint that upper bounds sender's sampling rate. Such a constraint is reasonable in practice since many real-world information senders (e.g., an IoT sensor) have limited battery, and thus cannot support a high sampling rate. Here we also remark that for our problem,

even when the task does not give R_u , D shall be an upper bound on the throughput D/T since $T \geq 1$.

Given a solution f , we denote the aggregate amount of data sent to link $e \in E$ at the offset $i \in \{0, 1, \dots, \mathcal{T}(f) - 1\}$, or equivalently the aggregate amount of data sent to e at each time $k \cdot \mathcal{T}(f) + i, \forall k \in \mathbb{Z}, k \geq 0$, 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 $0 \leq i \leq \mathcal{T}(f) - 1$, because $x_e(i + \mathcal{T}(f))$ is equal to $x_e(i)$ considering that f is periodically repeated. The link bandwidth constraints require $x_e(i)$ to be no greater than b_e , i.e., $x_e(i) \leq b_e$, for any link $e \in E$ and any offset $i = 0, 1, \dots, \mathcal{T}(f) - 1$. 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 $k \in \mathbb{Z}, k \geq 0$ such that $k \cdot \mathcal{T}(f) + \mathcal{B}^p(\vec{u}, e) = i$. Therefore, our **link bandwidth constraints** are

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

Suppose Λ_p^R (resp. Λ_a^R) is the minimal peak Aol (resp. minimal average Aol) 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 $r \in V \setminus \{s\}$, throughput requirements R_l and R_u , in this paper we are interested in the following two Aol 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 Aol, i.e.,

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

and obtain the feasible periodically repeated solution which has a throughput of R_p and a peak Aol of $\Lambda_p^{R_p}$. We denote this problem of Minimizing Peak Aol 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 Aol, i.e.,

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

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

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

As presented in this section, we consider a stylized model to analyze Aol. Our stylized model is deterministic, and hence ignores randomnesses that exist in many real world networks

(e.g., randomnesses of packet generation, network connectivity, and packet delays). Our study focuses on theoretically understanding the Aol for multi-path routing optimization in a multi-hop network. This problem is important but has not been studied, to our best knowledge. Therefore, we believe our study is of interest to the Aol research community. We leave it as a future direction to investigate a more practical network communication model that takes randomness into account.

C. Remark on Our Constant Link Delay Model

In this paper we consider a constant link delay model: 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 to it, and this data experiences a delay of d_e slots to pass it. Here we remark that this constant delay model is practically useful.

It is well-known that the networking delay is mainly composed of propagation delay, transmission delay, and queuing delay. As discussed in [46], the constant link delay model can take all three kinds of delays into consideration. Specifically,

- 1) Link propagation delay. It is the time for a signal to pass the link, and can be computed as the ratio between the link length and the propagation speed. Clearly it is a constant for each $e \in E$ and can be counted into d_e .
- 2) Link transmission delay. It is the time required to push the data onto the link. In general, it is S_e/b_e where S_e is the size of the data required to be pushed onto the link $e = (v, w) \in E$ at time t . Our constant link delay model can consider this link transmission delay in a discretized way: If $S_e \leq b_e$, the incurred link transmission delay is $\lceil S_e/b_e \rceil = 1$ and is counted into d_e . Otherwise if $b_e < S_e \leq 2b_e$, the link bandwidth constraint restricts that e first pushes b_e amount of data onto it at time t , experiencing a transmission delay of 1 that is counted into d_e . Then the remaining $S_e - b_e$ amount of data will be held at the node v for one time slot, and is pushed to e at time $t + 1$, experiencing a transmission delay of 2, where 1 delay is counted into d_e , and the remaining 1 delay is incurred by holding the data at node v from time t to time $t + 1$. Similarly, our constant link delay model can figure out the transmission delay for arbitrary $S_e : (k - 1)b_e < S_e \leq kb_e, \forall k \in \mathbb{Z}^+$.
- 3) Node queuing delay. It is the time the data spends in the routing queue of a node. When data arrives at a router (node), it has to be processed and then transmitted. But due to the finite service rate λ_v of a router $v \in V$, the router puts the data into the queue until it can get around to transmitting them if the data arrival rate is larger than λ_v . If we want to consider the queuing delay of a router node v , (i) we add a new node v' to V and change all the outgoing links of v to be the outgoing links of v' ; and (ii) we add a new link (v, v') to E , set its bandwidth to be λ_v and delay to be 1. Then the queuing delay of the router v in the original network can be given by the transmission delay of link (v, v') in the updated network, which can be taken into consideration by our constant link delay model in a discretized way.

As illustrated above, our constant link delay model can consider a special kind of traffic-dependent delay, i.e., S_e/b_e

where S_e is the link traffic load, in a discretized way. It may not model other traffic-dependent delays. To our best knowledge, this work presents the first step to study the Aol minimization problem of optimizing multi-path routing. The simple constant link delay model helps us obtain a theoretically complete understanding on this fundamental problem. It is an important future direction to consider more practical link delay models, e.g., model link delay as a convex non-decreasing function of link traffic. Note that problem becomes much more difficult if we model the link delay as a function of link traffic than the current setting. As an example, to minimize the maximum delay, it is APX-hard (hence no Polynomial-Time Approximation Scheme, or PTAS in short, exists unless $P = NP$) if the link delay is a traffic-dependent function [47], but a PTAS exists if the link delay is a constant [2].

IV. COMPARE Aol WITH MAXIMUM DELAY

In this section, we compare the problem of minimizing Aol (MPA and MAA) with that of minimizing maximum delay (MMD) theoretically under the single-unicast setting.

Consider a network with nodes s and r , and one link (s, r) . Suppose the delay (resp. bandwidth) of the link is d (resp. $b \geq D$). 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 is feasible and can have a task activation period of $T \leq T_u$. And the batch of data of the period starting at time $k \cdot T$ will be received by r at time $k \cdot T + d$. Now consider two different task activation periods T_1 and T_2 with $T_1 < T_2 \leq T_u$. From the perspective of minimizing maximum delay, the solution with $T = T_1$ is equivalent to that with $T = T_2$, because they obtain the same maximum delay of d . From the perspective of minimizing peak/average Aol, in contrast, the solution with $T = T_1$ is better than that with $T = T_2$, since according to Lemma 1 introduced later, the peak Aol (resp. average Aol) 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, but the throughput achieved by T_1 (D/T_1) is greater than that achieved by T_2 (D/T_2).

From the above example we observe that for periodic transmission services, Aol, instead of maximum delay, should be optimized. This is mainly because Aol is a 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 Aol, but it must be with bounded optimality loss compared to optimal.

Given a solution f , first we give a lemma to mathematically relates the Aol of f to the maximum delay of f .

Lemma 1: In the single-unicast setting, 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 Part B of supplementary materials. ■

A direct corollary is that the peak Aol (resp. average Aol) 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 Section II, only special cases of MMD with $R_l = R_u$, i.e., the quickest flow problem ($R_l = R_u \rightarrow 0$) and the min-max-delay flow problem ($R_l = R_u = D$), have been studied in the literature, and it is not clear how to solve MMD even given $0 < R_l = R_u \leq D$. Moreover, for general settings with $R_l < R_u$, we observe that both MPA and MAA can differ from MMD as follows.

Lemma 2: In the single-unicast setting, 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 Aol (resp. average Aol, maximum delay) of this instance. The following must hold for this instance

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

Further, there must exist an instance satisfying that

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

Proof: Refer to Part C of supplementary materials. ■

In Lemma 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 Aol. Similarly, we define $\vec{R}_a$ and $\vec{R}_m$ both as sets of throughputs.

Lemma 2 suggests that (i) minimizing maximum delay can differ from minimizing Aol, because the maximum-delay-optimal solution can achieve suboptimal peak/average Aol. (ii) The throughput of the maximum-delay-optimal solution must be no greater than that of the peak-/average- Aol-optimal solution. In the following, we further characterize near-tight optimality losses for the suboptimal Aol achieved by the maximum-delay-optimal solution.

Lemma 3: In the single-unicast setting, 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 Aol (resp. average Aol, maximum delay) of this instance. The following must hold for this instance

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

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

The peak Aol gap is near-tight, in the sense that for arbitrary D , R_l , and R_u with $D > 0$, $D/R_l \in \mathbb{Z}^+$, and $D/R_u \in \mathbb{Z}^+$, there is an instance satisfying the following

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

The average Aol gap is near-tight, in a similar sense with the following held

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

Proof: Refer to Part D of supplementary materials. ■

Overall, we observe that MPA and MAA are non-trivial as compared to MMD: (i) Aol-optimal solution, instead of maximum-delay-optimal one, is the time-critical solution for periodic transmission services; (ii) Aol-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 Aol-optimal solution is also maximum-delay-optimal, and vice versa, existing maximum delay minimization studies have strong assumptions on the fixed throughput (either $R_l = R_u \rightarrow 0$ or $R_l = R_u = D$). It is not clear how to minimize maximum delay with the throughput fixed arbitrarily ($0 < R_l = R_u \leq D$). In the following two sections, we design an optimal algorithm and an approximation framework for MPA and MAA under the single-unicast setting.

V. PROBLEM STRUCTURES OF MPA AND MAA

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

A. MPA is Different from MAA

Lemma 4: In the single-unicast setting, given any instance of MPA (or MAA), suppose $\vec{R}_p$ (resp. $\vec{R}_a$) is the optimal set of throughputs that minimize peak Aol (resp. average Aol) of this instance. For this instance,

- 1) the following must hold

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

- 2) and we have the following for any $R_p \in \vec{R}_p$ and any $R_a \in \vec{R}_a$

$$\begin{aligned} \Lambda_p^{R_p} - \Lambda_a^{R_a} &\leq \frac{D}{2R_l} - \frac{D}{2R_u}, \\ \Lambda_p^{R_a} - \Lambda_p^{R_p} &\leq \left\lfloor \frac{D}{2R_l} - \frac{D}{2R_u} \right\rfloor. \end{aligned}$$

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 Part E of supplementary materials. ■

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

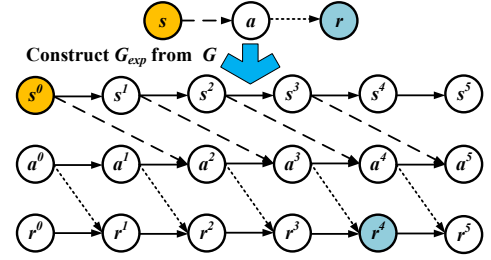


Fig. 2. An example of constructing G_{exp} . Suppose $d(s, a) = 2$ and $d(a, r) = 1$ in G . And suppose $M_U = 5$ and $M = 4$.

B. Both MPA and MAA are NP-Hard

Although MPA differs from MAA, we observe that both of them are NP-hard, because (i) based on Lemma 1, MMD given $R_l = R_u = D$ is a special case of MPA and MAA, and (ii) as discussed in Section II, the min-max-delay flow problem under our setting is exactly the problem MMD given $R_l = R_u = D$, which has been proven to be NP-hard by [2].

Lemma 5: Both MPA and MAA are NP-hard.

Proof: The min-max-delay flow problem which has been proven to be NP-hard in [2] is not only a special case of MPA but also a special case of MAA. ■

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-Aol-optimal R_p (resp. average-Aol-optimal R_a), together with the optimal periodically repeated solution.

C. Pseudo-Polynomial Time Algorithm for Obtaining $\mathcal{M}^R$

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 Lemma 1, the minimal peak Aol Λ_p^R and the minimal average Aol Λ_a^R can be achieved by the same solution.

Construct an expanded network. We construct an expanded network $G_{\text{exp}}(V_{\text{exp}}, E_{\text{exp}})$, 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 $v^i, i = 0, \dots, M_U$ (the loop in line 5). By this expansion, node v^i represents the node v at time $k \cdot T + i$ from the perspective of the period starting at time $k \cdot T$, where $T = D/R$. Second we expand each link $e = (v, w) \in E$ to links $(v^i, w^{i+d_e}), i = 0, \dots, M_U - d_e$ (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 $k \cdot T + i$ from the perspective of the period starting at time $k \cdot T$, where $T = D/R$. Third, we add links $(v^i, v^{i+1}), i = 0, \dots, M_U - 1$ (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 $M \leq M_U$, we observe that the problem of whether there exists a feasible periodically repeated solution

Algorithm 1 Construct G_{exp} from G

```

1: input:  $G = (V, E)$ ,  $M_U$ 
2: output:  $G_{\text{exp}} = (V_{\text{exp}}, E_{\text{exp}})$ 
3: procedure
4:    $V_{\text{exp}} = E_{\text{exp}} = \text{NULL}$ 
5:   for  $v \in V$  and  $i = 0, 1, \dots, M_U$  do
6:     Push node  $v^i$  into  $V_{\text{exp}}$ 
7:     for  $e = (v, w) \in E$  and  $i = 0, 1, \dots, M_U - d_e$  do
8:       Push link  $(v^i, w^{i+d_e})$  into  $E_{\text{exp}}$ 
9:     for  $v \in V$  and  $i = 0, 1, \dots, M_U - 1$  do
10:      Push link  $(v^i, v^{i+1})$  into  $E_{\text{exp}}$ 
11:   return  $G_{\text{exp}} = (V_{\text{exp}}, E_{\text{exp}})$ 

```

f in G , with $\mathcal{T}(f) = D/R$ and $\mathcal{M}(f) \leq M$, can be solved by solving a network flow problem that is casted by the following linear program in G_{exp} :

$$\max \sum_{e' \in \text{Out}(s^0)} x_{e'} \quad (2a)$$

$$\text{s.t.} \quad \sum_{e' \in \text{Out}(s^0)} x_{e'} = \sum_{e' \in \text{In}(r^M)} x_{e'}, \quad (2b)$$

$$\sum_{e' \in \text{Out}(v)} x_{e'} = \sum_{e' \in \text{In}(v)} x_{e'}, \forall v \in V_{\text{exp}} \setminus \{s^0, r^M\}, \quad (2c)$$

$$\sum_{e' \in e(i)} x_{e'} \leq b_e, \forall e \in E, \forall i = 0, \dots, D/R - 1, \quad (2d)$$

$$\text{vars. } x_{e'} \geq 0, \forall e' \in E_{\text{exp}}. \quad (2e)$$

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 Section III-B. 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 (2a) maximizes the amount of data sent from the sender of each period. Constraint (2b) restricts those data arrive at the receiver no later than M time slots as compared to the starting time of the period. Constraints (2c) are flow conservation constraints, and constraints (2d) are link bandwidth constraints.

Lemma 6: In the single-unicast setting, 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 $\mathcal{M}(f) \leq M$ is feasible if and only if the value of the optimal solution to the linear program (2) is no smaller than D .

Proof: Refer to Part F of supplementary materials. ■

To obtain $\mathcal{M}^R$, Lemma 6 suggests that we can use binary search to obtain the minimal integer $M^* \in [0, M_U]$, under

which the linear program (2) outputs a feasible flow with a value no smaller than D , and it is clear that this M^* shall be $\mathcal{M}^R$ (see Algorithm 2). Note that to construct G_{exp} , we need a $M_U \geq \mathcal{M}^R$. 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 $\vec{u} \in \vec{U}$ 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 lemma given in Part A of supplementary materials.

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

```

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

```

Lemma 7: Suppose $\mathcal{L}$ is the input size of the instance of linear program (2), then the time complexity of Algorithm 2 is $O(|E|^3 M_U^3 \mathcal{L} \log M_U)$.

Proof: Refer to Part G of supplementary materials. ■

Lemma 7 shows Algorithm 2 has a pseudo-polynomial time complexity: Algorithm 2 relies on solving a maximum flow problem in G_{exp} (Line 8), which is known to be solvable in a time polynomial with the size of G_{exp} .⁴ Hence Algorithm 2 has a time complexity polynomial with the size of G_{exp} . The size of G_{exp} is pseudo-polynomial with problem inputs: (i) considering $M_U \leq |V| \cdot (d_{\max} + D/R_l)$, the 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 .

D. Use Algorithm 2 to Solve MPA and MAA Optimally

Lemma 8: Both Λ_p^R and Λ_a^R are non-monotonic, non-convex, and non-concave with R theoretically.

Proof: Refer to Part H of supplementary materials. ■

Lemma 8 suggests that to solve MPA (resp. MAA) optimally, we need to enumerate Λ_p^R (resp. Λ_a^R) for all $R \in$

⁴It is clear that the size of the linear program (2) is polynomial with the size of the graph G_{exp} . The optimal solution of the program is a flow defined on links. After a flow decomposition [48], then we can get a flow defined on paths. Note that the time complexity of flow decomposition is polynomial with the size of the graph G_{exp} [48].

$[R_l, R_u]$, $D/R \in \mathbb{Z}^+$, and obtain the optimal one that achieves minimal peak Aol (resp. minimal average Aol). 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.

By Lemma 7, 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 1: In the single-unicast setting, both MPA and MAA are NP-hard in the weak sense.

Proof: A direct result of Lemma 5 and Algorithm 2. ■

VI. AN APPROXIMATION FRAMEWORK

As discussed in Section IV, the peak/average Aol 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 Aol. However, this idea is non-trivial, considering that as discussed in Section II, 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 Aol 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 under the single-unicast setting, 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 MPA and MAA approximately in a polynomial time.

A. MMD1: Maximum-Delay-Based Counterpart of Aol-Aware Problems MPA and MAA

We denote a feasible instance of MPA (resp. MAA) characterized by (G, s, r, R_l, R_u, D) as $\text{MPA}(R_l, R_u, D)$ (resp. $\text{MAA}(R_l, R_u, D)$). We denote the min-max-delay flow problem instance as $\text{MMD1}(R)$, which is defined by the same G, s, r , but with a throughput requirement of R .

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 [2]), 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, $\mathbf{f}$ is a special case of f . Furthermore, we have the following lemma comparing the min-max-delay flow problem with MPA and MAA.

Lemma 9: In the single-unicast setting, given any $\text{MPA}(R_l, R_u, D)$ (resp. $\text{MAA}(R_l, R_u, D)$), suppose $R \in [R_l, R_u]$, $D/R \in \mathbb{Z}^+$ is an arbitrary feasible throughput for it. Then $\text{MMD1}(R)$ must be feasible. Moreover, suppose $\mathbf{f}(R)$ is an arbitrary feasible solution to $\text{MMD1}(R)$, it holds that $\mathbf{f}(R)$ must be a feasible periodically repeated solution to $\text{MPA}(R_l, R_u, D)$ (resp. $\text{MAA}(R_l, R_u, D)$) with the following

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

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

Proof: Refer to Part I of supplementary materials. ■

Lemma 9 suggests that any feasible solution to the min-max-delay flow problem achieving a throughput of R is a feasible periodically repeated solution to 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 Aol (resp. average Aol) can be strictly greater than the minimal peak Aol (resp. minimal average Aol) with a throughput of R , i.e., than Λ_p^R (resp. Λ_a^R). This is because a solution to the min-max-delay flow problem 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: 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.

B. Solve MPA and MAA Approximately by Solving MMD1

Lemma 9 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 $\text{MMD1}(R_1)$ is feasible, $\text{MMD1}(R_2)$ must be feasible given any $0 < R_2 \leq R_1$ (see the proof to the following theorem), a direct result of Lemma 9 is that R_l must be a feasible throughput for $\text{MPA}(R_l, R_u, D)$ (resp. $\text{MAA}(R_l, R_u, D)$). Therefore, it is clear that solving $\text{MMD1}(R_l)$ must output a feasible solution to $\text{MPA}(R_l, R_u, D)$ (resp. $\text{MAA}(R_l, R_u, D)$). In the following theorem, we further prove that any approximate solution to $\text{MMD1}(R_l)$ must be an approximate solution to $\text{MPA}(R_l, R_u, D)$ (resp. $\text{MAA}(R_l, R_u, D)$). For easier reference, we denote an arbitrary α -approximation algorithm of the min-max-delay flow problem as $\text{ALG-MMD1}(\alpha)$.

Theorem 2: In the single-unicast setting, given any $\text{MPA}(R_l, R_u, D)$ and $\text{MAA}(R_l, R_u, D)$ where $D/R_l \in \mathbb{Z}^+$, $D/R_u \in \mathbb{Z}^+$, suppose we use $\text{ALG-MMD1}(\alpha)$ to solve the corresponding $\text{MMD1}(R_l)$. Then it must give an α -approximate solution $\mathbf{f}_\alpha(R_l)$ to $\text{MMD1}(R_l)$. Moreover, $\mathbf{f}_\alpha(R_l)$ must be a feasible periodically repeated solution to $\text{MPA}(R_l, R_u, D)$ and $\text{MAA}(R_l, R_u, D)$, with an approximation ratio of $(\alpha+c)$ where c is defined below

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

Proof: Refer to Part J of supplementary materials. ■

Theorem 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 . Theorem 2 also shows that this approximation framework for MPA and MAA has the same time complexity as the used algorithm for the min-max-delay flow problem. Note that polynomial time approximation algorithms for the min-max-delay flow problem exist in the literature, e.g., Misra *et al.* [2] have designed a $(1 + \epsilon)$ -approximation algorithm with a polynomial time complexity, where ϵ can be an arbitrary user-defined positive number.

VII. EXTENSION TO MULTIPLE-UNICAST SETTING

A. Multiple-Unicast Problem Definition

To extend our results from the single-unicast scenario to the multiple-unicast scenario, we first generalize definitions given in Sections III-A and III-B to the multiple-unicast setting.

Under the multiple-unicast setting, we assume there are K users, where each user sends a batch of data periodically from a sender to a receiver. In particular, for each user $i \in \{1, 2, \dots, K\}$, given that its batch generation period is T_i ($T_i \in \mathbb{Z}^+$) and batch generation offset is o_i ($o_i \in \mathbb{Z}, o_i \in [0, T_i - 1]$), there will be D_i ($D_i \in \mathbb{R}$) amount of data generated at a sender $s_i \in V$ at time $k \cdot T_i + o_i$ for each $k \in \mathbb{Z}, k \geq 0$, and this batch of data is required to be sent to a receiver $r_i \in V \setminus \{s_i\}$, possibly using multiple paths. We denote $\langle s, r \rangle$ as $\{\langle s_i, r_i \rangle, i = 1, \dots, K\}$, and denote $\vec{D}$ as $\{D_i, i = 1, \dots, K\}$.

We aim to obtain a multiple-unicast periodically repeated solution $f \triangleq \{f_i, i = 1, 2, \dots, K\}$, where each f_i is a single-unicast periodically repeated solution that is defined in our Section III-A and sends the batch of data of user i periodically from s_i to r_i . Based on the same definitions in Sections III-A and III-B, we can define the maximum delay, peak Aol, and average Aol of each f_i , respectively.

Under the multiple-unicast setting, let us denote $\mathcal{T}(f_i)$ as the batch generation period of f_i , and $\mathcal{O}(f_i)$ as the batch generation offset of f_i . The minimum (resp. maximum) throughput requirement requires each f_i to send D_i amount of data every $\mathcal{T}(f_i) \in \mathbb{Z}^+$ slots, achieving a throughput no smaller than an input $R_i^l \in \mathbb{R}$ (resp. no greater than an input $R_i^u \in \mathbb{R}$). The link bandwidth constraints require that the aggregate amount of data sent to each link $e \in E$ cannot be greater than its bandwidth b_e at each slot. For each f_i , we denote its amount of data sent to $e \in E$ at the offset $j \in \{0, 1, \dots, \mathcal{T}(f_i) - 1\}$, or its amount of data sent to e at each time $k \cdot \mathcal{T}(f_i) + \mathcal{O}(f_i) + j$ equivalently, as $x_e(f_i, j)$. Now suppose $\mathcal{T}(f)$ is a common multiple of $\{\mathcal{T}(f_1), \mathcal{T}(f_2), \dots, \mathcal{T}(f_K)\}$, i.e., suppose $\mathcal{T}(f) = m_i \cdot \mathcal{T}(f_i), m_i \in \mathbb{Z}^+$ for each $i \in \{1, 2, \dots, K\}$, the link bandwidth constraints are

$$\sum_{i=1}^K x_e(f_i, n_{i,j}) \leq b_e, \quad \forall j = 0, 1, \dots, \mathcal{T}(f) - 1, \forall e \in E,$$

where $n_{i,j}$ is defined as

$$n_{i,j} = \begin{cases} j\% \mathcal{T}(f_i) - \mathcal{O}(f_i), & \text{if } j\% \mathcal{T}(f_i) \geq \mathcal{O}(f_i), \\ j\% \mathcal{T}(f_i) + \mathcal{T}(f_i) - \mathcal{O}(f_i), & \text{otherwise.} \end{cases}$$

Under the multiple-unicast setting, our Aol minimization problems find a periodically repeated solution f either to minimize the weighted summation of peak Aol (i.e., $\min \sum_{i=1}^K w_i \cdot \Lambda_p(f_i)$), or to minimize the weighted summation of average Aol (i.e., $\min \sum_{i=1}^K w_i \cdot \Lambda_a(f_i)$), subject to throughput requirements and link bandwidth constraints. Here $w_i \geq 0$ is a weight imposed on the achieved Aol of the user i .

In Section VI, we designed an approximation framework for minimizing Aol in the single-unicast setting. In the following we generalize it to the multiple-unicast setting.

B. MMD1: Maximum-Delay-Based Counterpart of Aol-Aware Problems MPA and MAA

Denote $\text{MMD1}(\vec{R})$ as the maximum-delay-based counterpart of MPA and MAA in the multiple-unicast setting: Given K users each of which is associated with a sender-receiver pair $\langle s_i, r_i \rangle$ and a throughput requirement R_i , $\text{MMD1}(\vec{R})$ finds a set of paths from s_i to r_i for each user i , such that the aggregate allocated bandwidth of the set of paths is equal to R_i for each user i , the weighted summation of maximum delays of all users is minimized, and the link bandwidth constraints are satisfied. Notice that $\text{MMD1}(\vec{R})$ is NP-hard, with polynomial-time approximation algorithms proposed in [49].

Comparing MMD1 with our MPA (resp. MAA), we observe that given a throughput vector $\vec{R} = \{R_i, i = 1, 2, \dots, K\}$ with $R_i^l \leq R_i \leq R_i^u, \forall i = 1, 2, \dots, K$, $\vec{R}$ is feasible to MMD1 if and only if it is feasible to MPA (resp. MAA). This observation is based on a similar analysis discussed in Section VI.

Let us denote a feasible instance of MPA (resp. MAA) characterized by $(G, \langle s, r \rangle, \vec{R}_l, \vec{R}_u, \vec{D})$ as $\text{MPA}(\vec{R}_l, \vec{R}_u, \vec{D})$ (resp. $\text{MAA}(\vec{R}_l, \vec{R}_u, \vec{D})$). We have the following lemma.

Lemma 10: In the multiple-unicast setting, given any instance of $\text{MPA}(\vec{R}_l, \vec{R}_u, \vec{D})$ (resp. $\text{MAA}(\vec{R}_l, \vec{R}_u, \vec{D})$), suppose the throughput vector $\vec{R} = \{R_i, i = 1, 2, \dots, K\}$ which satisfies the following

$$R_i^l \leq R_i \leq R_i^u \text{ and } D_i/R_i \in \mathbb{Z}^+, \quad \forall i = 1, 2, \dots, K,$$

is feasible to this instance. Then $\text{MMD1}(\vec{R})$ must be feasible. Moreover, suppose $\mathbf{f}(\vec{R}) = \{\mathbf{f}_1(R_1), \dots, \mathbf{f}_K(R_K)\}$ is an arbitrary feasible solution to $\text{MMD1}(\vec{R})$, it holds that $\mathbf{f}(\vec{R})$ must be a feasible periodically repeated solution to $\text{MPA}(\vec{R}_l, \vec{R}_u, \vec{D})$ (resp. $\text{MAA}(\vec{R}_l, \vec{R}_u, \vec{D})$) with the following

$$\mathcal{M}(\mathbf{f}_i(R_i)) = \hat{\mathcal{M}}(\mathbf{f}_i(R_i)) + D_i/R_i - 1, \quad \forall i = 1, 2, \dots, K.$$

$\hat{\mathcal{M}}(\mathbf{f}_i)$ is the maximum delay of $\mathbf{f}_i$ with $\text{MMD1}(\vec{R})$.

Proof: Refer to Part K of supplementary materials. ■

Lemma 10 generalizes our Lemma 9 from the single-unicast setting to the multiple-unicast setting. Lemma 10 suggests that any feasible solution $\mathbf{f}(\vec{R})$ to $\text{MMD1}(\vec{R})$ is a feasible periodically repeated solution to the corresponding $\text{MPA}(\vec{R}_l, \vec{R}_u, \vec{D})$ (resp. $\text{MAA}(\vec{R}_l, \vec{R}_u, \vec{D})$), achieving a throughput of $\vec{R}$.

C. Solve MPA and MAA Approximately by Solving MMD1

Given any $\vec{R}^1 = \{R_i^1, i = 1, 2, \dots, K\}$ and $\vec{R}^2 = \{R_i^2, i = 1, 2, \dots, K\}$ with $R_i^2 \leq R_i^1, \forall i = 1, 2, \dots, K$, it is easy to prove that if $\text{MMD1}(\vec{R}^1)$ is feasible, $\text{MMD1}(\vec{R}^2)$ must be feasible. Thus $\vec{R}_l$ must be a feasible throughput for $\text{MPA}(\vec{R}_l, \vec{R}_u, \vec{D})$ (resp. $\text{MAA}(\vec{R}_l, \vec{R}_u, \vec{D})$) according to Lemma 10. Hence we propose to solve $\text{MMD1}(\vec{R}_l)$ in order to provide approximate solutions to our $\text{MPA}(\vec{R}_l, \vec{R}_u, \vec{D})$ (resp. $\text{MAA}(\vec{R}_l, \vec{R}_u, \vec{D})$).

Theorem 3: In the multiple-unicast setting, given any instance of $\text{MPA}(\vec{R}_l, \vec{R}_u, \vec{D})$ and $\text{MAA}(\vec{R}_l, \vec{R}_u, \vec{D})$ where $D_i/R_i^l \in \mathbb{Z}^+$ and $D_i/R_i^u \in \mathbb{Z}^+$ for each $i = 1, 2, \dots, K$, then $\text{MMD1}(\vec{R}_l)$ must be feasible. Moreover, if $\mathbf{f}_\alpha(\vec{R}_l)$ is an α -approximate solution to $\text{MMD1}(\vec{R}_l)$, it must be a feasible periodically repeated solution to $\text{MPA}(\vec{R}_l, \vec{R}_u, \vec{D})$ and $\text{MAA}(\vec{R}_l, \vec{R}_u, \vec{D})$, with an approximation ratio of $(\alpha + c)$, where c is defined as

$$c = \begin{cases} 2 \cdot \frac{\sum_{i=1}^K w_i \cdot D_i / R_i^l}{\sum_{i=1}^K w_i \cdot D_i / R_i^u}, & \text{for } \text{MPA}(\vec{R}_l, \vec{R}_u, \vec{D}), \\ 3 \cdot \frac{\sum_{i=1}^K w_i \cdot D_i / R_i^l}{\sum_{i=1}^K w_i \cdot D_i / R_i^u}, & \text{for } \text{MAA}(\vec{R}_l, \vec{R}_u, \vec{D}). \end{cases}$$

Proof: Refer to Part L of supplementary materials. ■

Theorem 3 suggests that we can use any α -approximation algorithm of MMD1 to solve our MPA and MAA approximately instead, with a time complexity same as that of the used approximation algorithm, providing an approximation ratio that is a constant depending on α and inputs $\{w_i, D_i, R_i^l, R_i^u, i = 1, 2, \dots, K\}$.

D. Our Framework Can Solve Other Aol-Aware Problems

MPA and MAA minimize weighted summation of Aols. Another representative Aol-aware minimization objective is

$$\min_{i=1,2,\dots,K} \max_{f_i} (\Lambda_p(f_i)), \text{ or } \min_{i=1,2,\dots,K} \max_{f_i} (\Lambda_a(f_i)),$$

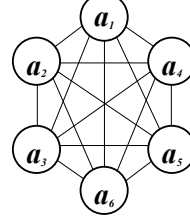
which minimizes worst peak/average Aol of individual users. Following the same proof to Lemma 10 and Theorem 3, it is easy to prove that the problem of minimizing worst peak/average Aol subject to user throughput requirements and link bandwidth constraints can be solved approximately by our proposed approximation framework, too. Overall, the design of our framework suggests a new avenue for developing approximation algorithms for minimizing Aol in the field of multi-path network communication.

VIII. PERFORMANCE EVALUATION

We evaluate the empirical performance of our proposed approaches in the single-unicast setting⁵, by simulating (i) two typical network topologies shown in Figure 3, and (ii)

⁵In this paper we do not simulate the multiple-unicast setting due to the following two concerns: (i) In the multiple-unicast setting we only have one approximation framework for optimizing Aol. There is no other algorithms for comparison. (ii) Our approximation framework uses existing algorithms of MMD1 to minimize Aol. To our best knowledge, only [49] solves MMD1 in the multiple-unicast setting in the literature. However, its performance is not satisfying, as it only supports part of the throughput requirement. We leave it as an important future direction of developing efficient algorithms for minimizing Aol in the multiple-unicast setting.

Complete Graph



Grid Graph

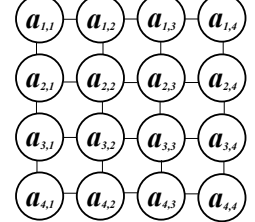
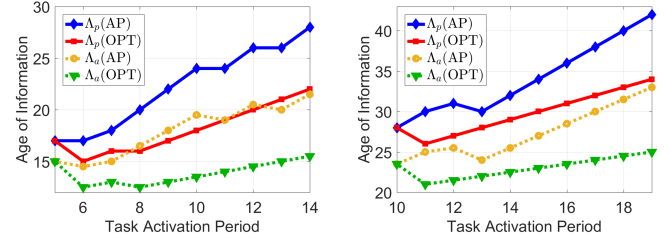


Fig. 3. Two simulated network topologies.



(a) An instance with $D = D_1$. (b) An instance with $D = D_2$.
Fig. 4. Results of two representative instances on the complete graph.

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 Lemma 9. 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 $\mathcal{T}(f) \in \{5, 6, \dots, 14\}$ (resp. $\mathcal{T}(f) \in \{10, 11, \dots, 19\}$) for simulations with $D = D_1$ (resp. with $D = D_2$). The $\text{ALG-MMD1}(\alpha)$ used by our approximation framework is the $(1 + \epsilon)$ -approximation algorithm from [2] 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 [50].

A. 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 the grid graph topology represents a distributed network structure. In Figure 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 $a_{4,4}$.

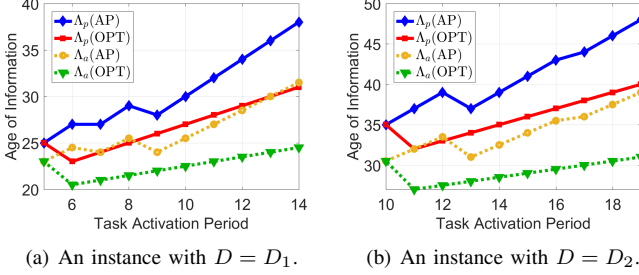


Fig. 5. Results of two representative instances on the grid graph.

First, for the complete network, we give the simulated Aol results of one representative instance with $D = D_1$ (resp. $D = D_2$) in Figure 4(a) (resp. Figure 4(b)). Similarly, for the grid network, we give the simulated Aol results of one representative instance with $D = R_1$ (resp. $D = R_2$) in Figure 5(a) (resp. Figure 5(b)). In all the figures, for each task activation period $T \in [T_l, T_u]$ ($T_l = D/R_u$ and $T_u = D/R_l$), $\Lambda_p(\text{AP})$ (resp. $\Lambda_a(\text{AP})$) is the peak Aol (resp. average Aol) of the solution of ALG-MMD1(2) with a throughput requirement of $R = D/T$, while $\Lambda_p(\text{OPT})$ (resp. $\Lambda_a(\text{OPT})$) is exactly the Λ_p^R (resp. Λ_a^R) with $R = D/T$, which is the peak Aol (resp. average Aol) of the solution given by our Algorithm 2. For the MPA (resp. MAA), note that the peak Aol (resp. average Aol) of the solution given by our approximation framework is the $\Lambda_p(\text{AP})$ (resp. $\Lambda_a(\text{AP})$) corresponding to the largest task activation period, i.e., with $T = T_u$, while the peak Aol (resp. average Aol) of the solution given by our pseudo-polynomial-time optimal algorithm is the minimal peak Aol (resp. minimal average Aol) in the figure.

From Figures 4 and 5, empirically we observe that (i) ALG-MMD1(2) outputs non-optimal peak Aol (resp. non-optimal average Aol) for any $T \in (T_l, T_u]$, as compared to Λ_p^R (resp. Λ_a^R) with $R = D/T$; (ii) Lemma 8 is verified by Figure 4(a), i.e., both the peak Aol and the average Aol are non-monotonic, non-convex, and non-concave with throughput; (iii) Lemma 4 is verified by Figure 4(a), where the task activation period of 6 (thus the throughput of $D_1/6$) achieving the optimal peak Aol is different from that of 8 (resp. that of $D_1/8$) achieving the optimal average Aol.

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 Aol results in average in Figure 6(a) (resp. Figure 6(b)). (i) Empirically, we observe that Λ_p^R and Λ_a^R are “almost” increasing with throughput R . (ii) Empirically, we observe that our optimal algorithm obtains a 3.8% peak Aol reduction (resp. 3.2% average Aol 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

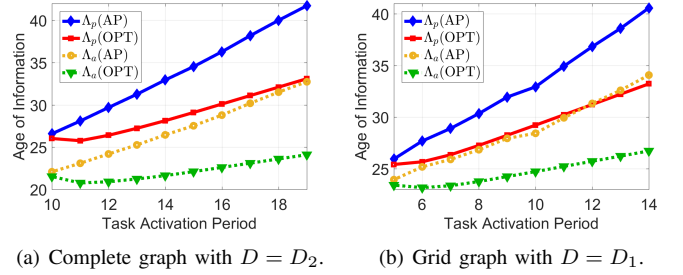


Fig. 6. Results in average of 200 instances on the typical graphs.

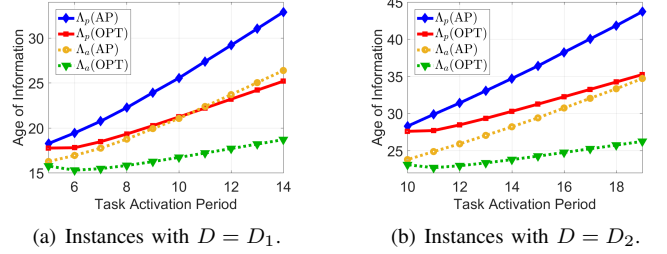


Fig. 7. Results in average of random graphs that are generated by SNAP [51].

increases by 0.10s when the number of possible throughputs increases by 1 (enumerate Aols achieved by all possible throughputs to figure out the optimal).

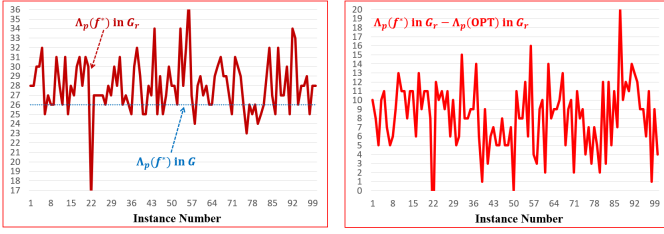
B. Simulations on Random Networks

We also use SNAP [51] 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 [51].

For each of the nine topologies, we run 100 simulation instances respectively with $D = D_1$ and with $D = D_2$. For each instance, the sender and the receiver are randomly selected. The simulated Aol results on random networks (Figure 7) is very similar to that on typical networks (Figure 6). When the range of task activation period of an instance increases by 1, (i) our optimal algorithm obtains a 4.3% peak Aol reduction (resp. 4.0% average Aol 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.

C. Simulations when the Network is not Deterministic

In this paper we consider a deterministic network model. However, for real-world networks, there may exist randomnesses (e.g., link delay may change with time). In the end of this section, we use simulations to give some insights on how our approach performs in a network that is not deterministic.



(a) Comparison between the Aol of f^* in G_r and the Aol of f^* in G . (b) Gap between the Aol of f^* in G_r and the optimal Aol in G_r .

Fig. 8. We consider 1 network G and 100 different networks G_r 's. $d_e = 3$ for each link e in G ; while d_e is randomly generated from $\{1, 2, 3, 4, 5\}$ for each link e in each G_r . f^* is the Aol-optimal solution in G . We illustrate the Aol performance of f^* in these 100 different G_r 's.

Let us consider the grid network in Figure 3, $D = 5 \cdot \mathcal{D}$, and $\mathcal{T}(f) \in \{5, 6, \dots, 14\}$. We first find the optimal solution f^* that minimizes Aol using our Algorithm 2 in a grid network $G(V, E)$ where $d_e = 3$ and $c_e = 30$ for each $e \in E$.⁶ Then we estimate the Aol performance of f^* in a grid network $G_r(V_r, E_r)$ where d_e is randomly generated from $\{1, 2, 3, 4, 5\}$ and $c_e = 30$ for each $e \in E_r$. We simulate 100 different G_r 's. For each of the 100 G_r 's, in Figure 8(a) we give the Aol of f^* , and in Figure 8(b) we give the difference between the Aol of f^* and the optimal (minimal) Aol.⁷

If there exists randomness for link delay, (i) from Figure 8(a) we observe that the Aol of f^* can be very different, which is expected. In average, the peak Aol of f^* is 28; (ii) from Figure 8(b) we observe the Aol of f^* is no longer optimal in most instances, with the average optimality gap to be 8. This gap is relatively small as compared to the Aol of f^* .

In this subsection we only simulate the setting when link delay can change. Clearly the performance of our approach will deteriorate when more randomnesses exist in the network, e.g., when link bandwidth can change. We leave it as a future direction to minimize Aol based on a more practical network communication model that takes randomness into account.

IX. CONCLUSION

We consider a single-unicast networking 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 Aol based on a stylized deterministic network model, by jointly optimizing (i) the throughput subject to throughput requirements, and (ii) the multi-path routing strategy. 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 Aol 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 min-

⁶In this setting the peak-Aol-minimal solution is the same as the average-Aol-minimal solution.

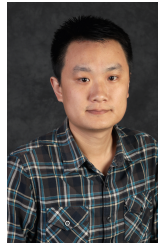
⁷In Figure 8(a), we do not give $\Lambda_a(f^*)$, because $\Lambda_p(f^*) - \Lambda_a(f^*)$ is a constant 2 in this setting. In Figure 8(b), we do not give $\Lambda_a(f^*) - \Lambda_a(\text{OPT})$, because it is the same as $\Lambda_p(f^*) - \Lambda_p(\text{OPT})$.

imization problem to solve our Aol minimization problems in a polynomial time, with an approximation ratio of $\alpha + c$. Here c is a constant depending on the throughput requirements. Furthermore, we show that our proposed polynomial-time approximation framework can be generalized to the multiple-unicast setting with strong theoretical performance guarantee. We conduct simulations over various network topologies to empirically validate the effectiveness of our approach.

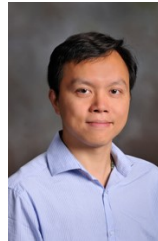
REFERENCES

- [1] Q. Liu, H. Zeng, and M. Chen, “Minimizing age-of-information with throughput requirements in multi-path network communication,” in *Proc. ACM Int'l Sym. Mobile Ad Hoc Networking and Computing*, 2019.
- [2] S. Misra, G. Xue, and D. Yang, “Polynomial time approximations for multi-path routing with bandwidth and delay constraints,” in *Proc. IEEE Int'l Conf. Computer Communications*, 2009.
- [3] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, “Minimizing age of information in vehicular networks,” in *Proc. IEEE Communications Society Conf. Sensor, Mesh and Ad Hoc Communications and Networks*, 2011.
- [4] S. Kaul, R. Yates, and M. Gruteser, “Real-time status: How often should one update?” in *Proc. IEEE Int'l Conf. Computer Communications*, 2012.
- [5] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff, “Update or wait: How to keep your data fresh,” *IEEE Trans. Information Theory*, vol. 63, no. 11, pp. 7492–7508, 2017.
- [6] B. T. Bacinoglu and E. Uysal-Biyikoglu, “Scheduling status updates to minimize age of information with an energy harvesting sensor,” in *Proc. IEEE Int'l Sym. Information Theory*, 2017.
- [7] Z. Zhou, C. Fu, C. J. Xue, and S. Han, “Transmit or discard: Optimizing data freshness in networked embedded systems with energy harvesting sources,” in *Proc. Design Automation Conf.*, 2019.
- [8] S. Feng and J. Yang, “Minimizing age of information for an energy harvesting source with updating failures,” in *Proc. IEEE Int'l Sym. Information Theory*, 2018.
- [9] A. Arafat, J. Yang, S. Ulukus, and H. V. Poor, “Age-minimal transmission for energy harvesting sensors with finite batteries: Online policies,” *IEEE Trans. Information Theory*, vol. 66, no. 1, pp. 534–556, 2019.
- [10] R. D. Yates, “Lazy is timely: Status updates by an energy harvesting source,” in *Proc. IEEE Int'l Sym. Information Theory*, 2015.
- [11] L. Huang and E. Modiano, “Optimizing age-of-information in a multi-class queueing system,” in *Proc. IEEE Int'l Sym. Information Theory*, 2015.
- [12] R. Talak, S. Karaman, and E. Modiano, “Can determinacy minimize age of information?” *arXiv preprint arXiv:1810.04371*, 2018.
- [13] I. Kadota, A. Sinha, and E. Modiano, “Optimizing age of information in wireless networks with throughput constraints,” in *Proc. IEEE Int'l Conf. Computer Communications*, 2018.
- [14] —, “Scheduling algorithms for optimizing age of information in wireless networks with throughput constraints,” *IEEE/ACM Trans. Networking*, vol. 27, no. 4, pp. 1359–1372, 2019.
- [15] Y.-P. Hsu, “Age of information: Whittle index for scheduling stochastic arrivals,” in *Proc. IEEE Int'l Sym. Information Theory*, 2018.
- [16] C. Li, S. Li, and Y. T. Hou, “A general model for minimizing age of information at network edge,” in *Proc. IEEE Int'l Conf. Computer Communications*, 2019.
- [17] C. Li, Y. Huang, Y. Chen, B. Jalaian, Y. T. Hou, and W. Lou, “Kronos: A 5G scheduler for AoI minimization under dynamic channel conditions,” in *Proc. IEEE Int'l Conf. Distributed Computing Systems*, 2019.
- [18] A. M. Bedewy, Y. Sun, S. Kompella, and N. B. Shroff, “Age-optimal sampling and transmission scheduling in multi-source systems,” in *Proc. ACM Int'l Sym. Mobile Ad Hoc Networking and Computing*, 2019.
- [19] R. Talak, S. Karaman, and E. Modiano, “Minimizing age-of-information in multi-hop wireless networks,” in *Proc. IEEE Allerton Conf. Communication, Control, and Computing*, 2017.
- [20] R. Talak, I. Kadota, S. Karaman, and E. Modiano, “Scheduling policies for age minimization in wireless networks with unknown channel state,” in *Proc. IEEE Int'l Sym. Information Theory*, 2018.
- [21] S. Farazi, A. G. Klein, and D. R. Brown, “Fundamental bounds on the age of information in multi-hop global status update networks,” *Journal of Communications and Networks*, vol. 21, no. 3, pp. 268–279, 2019.

- [22] R. Talak, S. Karaman, and E. Modiano, "Optimizing information freshness in wireless networks under general interference constraints," in *Proc. ACM Int'l Sym. Mobile Ad Hoc Networking and Computing*, 2018.
- [23] I. Kadota and E. Modiano, "Minimizing the age of information in wireless networks with stochastic arrivals," in *Proc. of ACM Int'l Sym. Mobile Ad Hoc Networking and Computing*, 2019.
- [24] X. Ran, H. Chen, Z. Liu, and J. Chen, "Delivering deep learning to mobile devices via offloading," in *ACM Workshop Virtual Reality and Augmented Reality Network*, 2017.
- [25] Y. Sun, "A collection of recent papers on the age of information," 2019, available at <http://webhome.auburn.edu/~yzs0078/>.
- [26] J. Wang, L. Li, S. H. Low, and J. C. Doyle, "Cross-layer optimization in tcp/ip networks," *IEEE/ACM Trans. Networking*, vol. 13, no. 3, pp. 582–595, 2005.
- [27] M. Wang, C. W. Tan, W. Xu, and A. Tang, "Cost of not splitting in routing: Characterization and estimation," *IEEE/ACM Trans. Networking*, vol. 19, no. 6, pp. 1849–1859, 2011.
- [28] O. Ayan, H. M. Gürsu, A. Papa, and W. Kellerer, "Probability analysis of age of information in multi-hop networks," *IEEE Networking Letters*, vol. 2, no. 2, pp. 76–80, 2020.
- [29] X. Zheng, S. Zhou, Z. Jiang, and Z. Niu, "Closed-form analysis of non-linear age of information in status updates with an energy harvesting transmitter," *IEEE Trans. Wireless Communications*, vol. 18, no. 8, pp. 4129–4142, 2019.
- [30] A. Maatouk, M. Assaad, and A. Ephremides, "On the age of information in a csma environment," *IEEE/ACM Trans. Networking*, vol. 28, no. 2, pp. 818–831, 2020.
- [31] C. Li, S. Li, Y. Chen, Y. T. Hou, and W. Lou, "AoI scheduling with maximum thresholds," in *Proc. IEEE Int'l Conf. Computer Communications*, 2020.
- [32] Q. Liu, C. Li, Y. T. Hou, W. Lou, and S. Kompella, "Aion: A bandwidth optimized scheduler with AoI guarantee," in *Proc. IEEE Int'l Conf. Computer Communications*, 2021.
- [33] G. D. Nguyen, S. Kompella, C. Kam, J. E. Wieselthier, and A. Ephremides, "Information freshness over an interference channel: A game theoretic view," in *Proc. IEEE Int'l Conf. Computer Communications*, 2018.
- [34] Z. Ning, P. Dong, X. Wang, X. Hu, L. Guo, B. Hu, Y. Guo, T. Qiu, and R. Kwok, "Mobile edge computing enabled 5G health monitoring for Internet of medical things: A decentralized game theoretic approach," *IEEE Journal on Selected Areas in Communications*, 2020.
- [35] P. Mayekar, P. Parag, and H. Tyagi, "Optimal source codes for timely updates," *IEEE Trans. Information Theory*, vol. 66, no. 6, pp. 3714–3731, 2020.
- [36] R. Devassy, G. Durisi, G. C. Ferrante, O. Simeone, and E. Uysal-Biyikoglu, "Delay and peak-age violation probability in short-packet transmissions," in *Proc. IEEE Int'l Sym. Information Theory*, 2018.
- [37] M. Lin and P. Jaillet, "On the quickest flow problem in dynamic networks: a parametric min-cost flow approach," in *Proc. ACM-SIAM Sym. Discrete algorithms*, 2015.
- [38] M. Saho and M. Shigeno, "Cancel-and-tighten algorithm for quickest flow problems," *Networks*, vol. 69, no. 2, pp. 179–188, 2017.
- [39] Q. Liu, L. Deng, H. Zeng, and M. Chen, "A tale of two metrics in network delay optimization," in *Proc. IEEE Int'l Conf. Computer Communications*, 2018.
- [40] —, "A tale of two metrics in network delay optimization," *IEEE/ACM Trans. Networking*, vol. 28, no. 3, pp. 1241–1254, 2020.
- [41] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," in *Proc. IEEE Conf. Decision and Control*, 1990.
- [42] A. Sinha and E. Modiano, "Optimal control for generalized network-flow problems," *IEEE/ACM Trans. Networking*, vol. 26, no. 1, pp. 506–519, 2017.
- [43] I.-H. Hou, V. Borkar, and P. Kumar, "A theory of qos for wireless," in *Proc. IEEE Int'l Conf. Computer Communications*, 2009.
- [44] I.-H. Hou and P. Kumar, "Utility maximization for delay constrained qos in wireless," in *Proc. IEEE Int'l Conf. Computer Communications*, 2010.
- [45] L. Deng, C.-C. Wang, M. Chen, and S. Zhao, "Timely wireless flows with general traffic patterns: Capacity region and scheduling algorithms," *IEEE/ACM Trans. Networking*, vol. 25, no. 6, pp. 3473–3486, 2017.
- [46] S. Bai, W. Zhang, G. Xue, J. Tang, and C. Wang, "Dear: Delay-bounded energy-constrained adaptive routing in wireless sensor networks," in *Proc. IEEE Int'l Conf. Computer Communications*, 2012.
- [47] J. R. Correa, A. S. Schulz, and N. E. Stier-Moses, "Fast, fair, and efficient flows in networks," *Operations Research*, vol. 55, no. 2, pp. 215–225, 2007.
- [48] T. Magnanti, R. Ahuja, and J. Orlin, "Network flows: theory, algorithms, and applications," *Prentice Hall, Upper Saddle River, NJ*, 1993.
- [49] Q. Liu, H. Zeng, and M. Chen, "Network utility maximization under maximum delay constraints and throughput requirements," *IEEE/ACM Trans. Networking*, vol. 28, no. 5, pp. 2132–2145, 2020.
- [50] IBM, "Cplex optimizer," 2017, available at <https://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/>.
- [51] J. Leskovec and R. Sosič, "Snap: A general-purpose network analysis and graph-mining library," *ACM Trans. Intelligent Systems and Technology*, vol. 8, no. 1, p. 1, 2016.



Qingyu Liu (Member, IEEE) is currently a Research Assistant Professor for Electrical and Computer Engineering with Virginia Tech. He received his Ph.D. in Computer Engineering from Virginia Tech in 2019, M.S. in Computer Science from Tsinghua University in 2014, and B.S. in Computer Science from Qingdao University in 2011. He was a visiting student in the Department of Information Engineering at The Chinese University of Hong Kong, from August 2017 to December 2017. He was a Postdoc Associate for Electrical and Computer Engineering with Virginia Tech from September 2019 to June 2021. His recent research interests include wireless and mobile networks, 5G/next-G, CPS/IoT, intelligent transportation, and optimization/algorithm design in networked systems.



Haibo Zeng (Member, IEEE) is with Department of Electrical and Computer Engineering at Virginia Tech. He received his Ph.D. in Electrical Engineering and Computer Sciences from University of California at Berkeley. He was a Senior Researcher at General Motors R&D until October 2011, and an Assistant Professor at McGill University until August 2014. His research interests are embedded systems, cyber-physical systems, and real-time systems. He received five best / outstanding paper awards in the above fields.



Minghua Chen (Fellow, IEEE) received his B.Eng. and M.S. degrees from the Department of Electronic Engineering, Tsinghua University. He received his Ph.D. degree from the Department of Electrical Engineering and Computer Sciences, University of California Berkeley. He is currently a Professor of School of Data Science, City University of Hong Kong. Minghua received the Eli Jury award from UC Berkeley in 2007 (presented to a graduate student or recent alumnus for outstanding achievement in the area of Systems, Communications, Control, or Signal Processing) and The Chinese University of Hong Kong Young Researcher Award in 2013. He also received several best paper awards, including IEEE ICME Best Paper Award in 2009, IEEE Transactions on Multimedia Prize Paper Award in 2009, ACM Multimedia Best Paper Award in 2012, and IEEE INFOCOM Best Poster Award in 2021. Minghua's recent research interests include online optimization and algorithm, machine learning in power system operation, intelligent transportation system, distributed optimization, delay-constrained network coding, and capitalizing the benefit of data-driven prediction in algorithm/system design. He is a Distinguished Member of ACM and a Fellow of IEEE.