MATHEMATICS OF OPERATIONS RESEARCH
Vol. 00, No. 0, Xxxxx 0000, pp. 000–000
ISSN 0364-765X | EISSN 1526-5471 | 00 | 0000 | 0001

INFORMS
DOI 10.1287/xxxx.0000.0000

Bounding Residence Times for Atomic Dynamic Routings

Zhigang Cao
School of Economics and Management, Beijing Jiaotong University, Beijing, China, zgcao@bjtu.edu.cn

Bo Chen
Warwick Business School, University of Warwick, Coventry, United Kingdom, b.chen@warwick.ac.uk

Xujin Chen
Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China, xchen@amss.ac.cn
School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing, China,

Changjun Wang*
Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China, wcj@amss.ac.cn
(Corresponding author)

In this paper, we are concerned with bounding agents' residence times in the network for a broad class of atomic dynamic routings. We explore novel token techniques to circumvent direct analysis on complicated chain effects of dynamic routing choices. Even though agents may enter the network over time for an infinite number of periods, we prove that under a mild condition, the residence time of every agent is upper bounded (by a network-dependent constant plus the total number of agents inside the network at the entry time of the agent). Applying this result to three game models of atomic dynamic routing in the recent literature, we establish that the residence times of selfish agents in a series-parallel network with a single origin-destination pair are upper bounded at equilibrium, provided the number of incoming agents at each time point does not exceed the network capacity (i.e., the smallest total capacity of edges in the network whose removal separates the origin from the destination).

Key words: atomic dynamic routing; residence time; token technique; selfish routing; Nash equilibrium
MSC2000 subject classification: Primary: 91A43; secondary: 91A50, 05C21
OR/MS subject classification: Primary: Games/group decisions: Noncooperative; secondary: Networks/graphs: Flow algorithms
History: Received January 14, 2021; revised September 20, 2021; accepted November 12, 2021.

1. Introduction.

Models of dynamic routing (also known as flows over time) find diverse applications in traffic flows and Internet data flows etc. They are recently drawing increasing research attention (see e.g., [6, 8, 12, 14, 19, 28, 29, 30, 33]). Dynamic routing models are usually classified into two types: atomic and non-atomic, with the former typically being more realistic (especially when steady states are hard to reach), and at the same time even more challenging to investigate. The difficulty stems from the fact that interactions among atomic agents could be formidably complicated due to their dynamic nature and hard-to-predict chain effects. In both atomic and non-atomic models, congestions are usually modeled by the deterministic queuing (DQ) rule [32], which roughly states that agents who cannot move forward due to edge capacity constraints have to queue and wait at the places where congestions happen. The DQ rule well simulates real traffic congestions and rests at the core of various agent-based simulation softwares, such as MATSim [16]. We use *queue length* to denote the number of waiting agents (resp. the amount of cumulated flows) at a place where congestion happens for atomic models (resp. nonatomic models). Queue lengths along with edge capacities determine agents' waiting times and at the same time are a measure of congestion level.

In dynamic routing games, selfish agents enter the network over time from their origins, and try to reach their destinations as soon as possible. An equilibrium is a state where no agent is incentivized to deviate unilaterally. A central task in studying these games is to understand the equilibria, most importantly their efficiencies. One of the questions is: when agents enter the network inexhaustibly for an infinite period of time, are the equilibrium queue lengths bounded by a finite number that is independent of the total amount of inflow? When all agents share the same origin and destination, by the well-known max-flow-min-cut theorem, it is natural to impose an additional assumption that the inflow rate, namely, the number of incoming atomic agents (in atomic routing) or the incoming rate of non-atomic agents (in non-atomic routing) at each time point does not exceed the network capacity. We refer to this question (with the additional assumption on inflow rate) as the *equilibrium queue length problem* (EQLP). The problem actually asks whether congestions can be arbitrarily bad at equilibrium. As an interesting open problem in the literature [6, 28], answering EQLP turns out to be a first step toward quantifying equilibrium efficiency of the corresponding dynamic routing games. It is easy to construct instances in which the queue lengths at equilibrium are network dependent (e.g., they can be as long as the longest paths in the network). So the finite upper bound in a possible affirmative answer to EQLP necessarily depends on network parameters in general. To the best of our knowledge, for most known models, EQLP can be equivalently rephrased as whether agents' residence times in the network at equilibrium are upper bounded.

In this paper, we study a broad class of atomic dynamic routings, which includes the models studied in [3, 12, 28] as special cases. This class is in essence the same as the packet routing model studied in [24]. In our model, the residence time an agent spends in the network is defined as in a usual DQ model: it is the sum of a free-flow transit time determined by the length of the agent's chosen origin-destination (OD) path and a total waiting (queuing) time caused by other agents. In contrast to usual DQ models, our model has no specific traffic regulations except that each edge should dispatch as many agents as possible and agents are not allowed to pause when they are able to move (a.k.a. greedy scheduling in [5, 24]). In particular, no specific tie-breaking rule is imposed when more agents than an edge can accommodate intend to move along it simultaneously (see Section 2 for a precise description).

1.1. Contributions. Our paper consists of two parts. Before studying the routing games in the second part, we analyze in the first part a more basic scenario of DQ-based atomic dynamic routing where no additional restriction is imposed on agents' behavior, whether strategic or not.

In the first part, we are concerned with bounding residence times of atomic agents. This problem is of interest in its own right and, at the same time, is the basis for analyzing other related problems including EQLP. Theorem 1 (see Section 3.1) provides a series of upper bounds for agents' residence times in unit-capacity networks. One particular result of Theorem 1 asserts that, under mild conditions, the residence time of any agent is upper bounded by the free-flow transit time spent on any one of the longest possible OD paths plus the number of agents that are inside the network when this agent enters. Theorem 2 (see Section 4) extends Theorem 1 to cases where networks have uniform capacities under an additional constraint that all agents enter the network at the very beginning. The above results are valid for a wide range of network structures: all acyclic networks with possibly multiple origins and multiple destinations. Our results also generalize the main result in [24] on shortest-path networks.

To establish these results, we develop two token-based techniques to circumvent direct analysis of complicated chain effects, which could contribute to future studies on atomic dynamic routing problems. The basic idea behind our techniques is that we switch our attention from agents to some imaginary objects called *tokens*. These tokens do not affect the dynamics of agents and they move more regularly than agents. Compared with agents, tokens are largely free of chain effects. On the other hand, the tokens' residence times are in some sense equivalent to the agents' residence times. We investigate both multi-token and single-token settings. In the multi-token setting, more than one token are applied and their movements are determined by the movements of the agents involved, such that there is a one-to-one mapping between the tokens and those agents involved. Whereas in the single-token setting, only one token is applied, but its movement is more flexible in that we can "control" it more freely to facilitate our analysis. It turns out that both techniques have their respective merits, the multi-token one being powerful for constantly incoming agents and the single-token one being more effective for uniform edge capacities.

It is well known that, compared with their non-atomic counterparts, very few (if any) general mathematical tools have been available to deal with atomic dynamic routing problems. Due to the generality of our model and the flexibility of token movements, we believe that our token technique and its variations have potential applications to more problems. While our multi-token technique is brand new, our single-token technique bears some resemblance to a method in [24], where the so-called train conductor could be regarded as the counterpart of our single token. Nevertheless, even for the single-token analysis, we have explored substantially new ideas, e.g., a critical partial order that not only provides new results on general networks but also greatly simplifies the proof of [24] on shortest-path networks (see Appendix B for more discussions).

In the second part of this paper, we focus on the EQLP for three classes of atomic dynamic routing games, studied in [3], [12] and [28], respectively, where competing agents queue according to agent priority list [12], or first-in-first-out principle with tie-breaking by agent priorities [28] or by edge priorities [3]. First, we show in Theorem 3 (see Section 5.2) that equilibria in the three game models possess two nice properties: agents entering the network earlier will not exit it later, and agents exiting the network later are unable to overtake those earlier at any intermediary vertex. These two properties, being of interest in their own, also imply that the condition in Theorem 1 is not restrictive. Second, based on Theorems 1 and 3, we provide an affirmative answer to EQLP in Theorem 4 (see Section 6.2) for the case where the network is series-parallel. To the best of our knowledge, this is the strongest result so far for the EQLP on atomic dynamic routing games. In particular, it improves upon a related result in [28] on chain-of-parallel networks, which are special cases of series-parallel networks. Since series-parallel networks can be recursively constructed, most problems in the literature related to this class of networks are studied through an inductive argument. However, substantial obstacles for a direct induction on the EQLP occur, because when a larger network satisfies the hypothesis of the EQLP, i.e., its inflow rate does not exceed its capacity, smaller sub-networks (from which the larger network is constructed) may well not satisfy this hypothesis. To overcome this difficulty, we explore a novel induction that utilizes a stronger inductive basis to establish a statement stronger than an affirmative answer to the EQLP on series-parallel networks.

1.2. Related literature. Our model follows the DQ rule, which is introduced in [32], developed in [13], and recently revivified in [18, 6, 28].

A line of research concerning the residence times of nonstrategic atomic agents investigates the store-and-forward packet routing problem for finding fast delivery of the packets in a unit-capacity network [22]. A fair amount of research in this area focuses on constructing a schedule for delivering the packets whose OD paths have been fixed, where the schedule specifies which packets move and which wait at each time step. In these studies, the most related routing rule to our work is the aforementioned greedy scheduling (that packets are forwarded whenever possible). Mansour and Patt-Shamir [24] study the greedy scheduling where packets take shortest paths. They prove that every packet i reaches its destination after at most $l_i + |\Delta| - 1$ steps, where l_i is the length of the shortest path that packet i takes and $|\Delta|$ is the number of packets in the network. When the used paths are not necessarily shortest, Cidon *et al.* [5] prove that for some very special networks, the residence time of packet i is at most the length of the route taken by packet i plus $|\Delta| - 1$. However, greedy scheduling might behave badly in general networks. The greedy scheduling studied in [24, 5] and in this paper is essentially a non-algorithmic rule, which does not require algorithmic coordinations among packets (agents). This stands in sharp contrast to other related work on packet routing, where centralized or distributed algorithms are needed to produce global or distributed arrangements for storing and forwarding packets collaboratively [1, 20, 21, 25, 27]. Leighton *et al.* [21, 20] prove that there exists a polynomial-time computable schedule under which the residence time of every packet is $O(l + b)$, where l is the length of the longest given path and b the largest number of packets that traverse a single edge during the entire course of the routing. When restricted to directed trees, Peis *et al.* [26] construct a schedule that guarantees the residence times to be bounded above by $l + b - 1$. For general networks, they show that if all packets take shortest paths and the lengths of these paths are pairwise different, then a schedule with residence times at most l can be found efficiently. Srinivasan and Teo [31] and Koch *et al.* [17] present algorithms that find OD paths of bounded l and b in general networks, leading to constant-factor approximations for the packet routing problem. Busch *et al.* [2] obtain better approximations for special network topologies: tree, mesh, butterfly, and hypercube. We refer the reader to [26] for more related literature, including study on the flow-over-time problem for finding fast non-atomic dynamic routings [9, 11, 15].

Graf and Harks [10] recently prove an upper bound on non-atomic agents' residence times for dynamic routing in an acyclic network with a single destination. Assuming all non-atomic agents enter the network during time interval $[0, \theta]$ and travel under greedy scheduling, the authors show that the routing terminates before time $\theta + l + \delta$, where l is the length of a longest OD path and δ is the total amount of flow during the routing.

For the EQLP on atomic dynamic routing games (with a single OD pair), the most related result to ours is given by Scarsini *et al.* [28], who show that (i) when agents queue under first-in-first-out principle and agent-priority tie-breaking rule, a special class of Nash equilibria (which they call *uniformly-fastest-route* equilibria and will also be investigated in this paper) exist, and (ii) queue lengths at these equilibria are indeed bounded, provided the underlying network is chain-of-parallel and the inflow (with rates no larger than the network capacity) is seasonal. The existence of a uniformly-fastest-route equilibrium when agents queue under agent-priority list [12] and that of a Nash equilibrium (NE) when agents queue under first-in-first-out principle and edge-priority tie-breaking rule [3] serve as bases of our study on the EQLP for the two classes of atomic dynamic routing games. The affirmative answer to the EQLP on series-parallel networks for the latter game was announced in [3] without a formal proof.

For non-atomic games with a single OD pair and a uniform inflow rate, the existence and uniqueness of the NE are proved by Cominetti *et al.* [6]. A recent breakthrough on the EQLP is obtained by Cominetti *et al.* [7]: Using a potential-function method, they show that given uniform inflow rate not exceeding the network capacity, the NE always reaches a steady state in finite time, after which queue lengths remain unchanged.

An affirmative answer to EQLP usually implies a bounded price of anarchy (PoA) and/or a bounded price of stability (PoS) for the game concerned, where the PoA (resp. PoS) is the ratio of the system objective value at a worst (resp. best) NE to that at an optimal solution. For atomic dynamic routing games where agents queue according to a given (global) agent-priority list, Harks *et al.* [12] analyze the efficiency of NEs for minimizing the total delay of all k agents who are ready to start their travels right from the beginning. They show that the asymmetric game (with multiple OD pairs) has a PoS of $\Omega(\sqrt{k})$ and a PoA upper bounded by $1+k^3/2$, while the symmetric game (with a single OD pair) has a PoS and a PoA equal to 1 and $(k+1)/2$, respectively.

For non-atomic dynamic routing games with a single OD pair and uniform inflow rate, Macko *et al.* [23] show that the PoA w.r.t. the min-max delay can be as large as $n-1$ in networks with n vertices. Correa *et al.* [8] study the PoA of these games under the system objective of minimizing the time required to route a given amount of flow from the origin to the destination. They show that, if some natural monotonicity conjecture holds (i.e., decreasing the inflow rate will increase the equilibrium makespan), then the PoA is exactly $e/(e-1)$.

The rest of the paper is organized as follows. Section 2 introduces our general routing model. Section 3 bounds agents' residence times for the general model on unit-capacity networks. Section 4 moves on to uniform-capacity networks. Section 5 introduces three game models under the general routing scheme as well as some basic concepts and properties on the equilibria. Section 6 studies the EQLP on series-parallel networks. Section 7 concludes this paper with some further discussions.

2. General routing model.

In this section, we introduce our general model of atomic dynamic routing (abbreviated as ADR).

2.1. Input network. We are given a finite directed multi-graph, henceforth called a network, $G=(V,E)$, where V is the vertex set and E the edge set. The network is assumed to be *acyclic*, unless otherwise stated. The network G may have multiple origin vertices and multiple destination vertices. For every vertex $v\in V$, we use E_v^+ and E_v^- to denote the set of outgoing edges from v and the set of incoming edges to v in G, respectively. All the paths discussed are directed and simple. Given a path P in G, the sub-path of P from vertex u to vertex v is written as $P[u,v]$. For vertex v with E_v^+ (resp. E_v^-) intersecting P, we use $\epsilon_v^+(P)$ (resp. $\epsilon_v^-(P)$) to denote the unique edge in the intersection, i.e., the edge of P outgoing from (resp. incoming to) v.

2.2. Atomic agents. The dynamic routing is represented by atomic agents moving through G, entering via the origins and exiting via the destinations. Time starts from 0. At each nonnegative integer time point, a (possibly empty) set of finitely many agents enter G from their origins. We use Δ to denote the set of all agents. Each agent $i\in\Delta$ moves along a path (called his *route* or *OD path*) from his origin o_i to his destination d_i. The routes of all agents, one for each agent, form a *path profile* of the ADR instance. We often identify a path profile of all agents with the *routing* determined by it, and also refer to it as an *ADR profile*.

2.3. Unit assumption. We assume that each edge of the input network G has a unit length and a unit capacity (more detailed discussion on general integer lengths and capacities can be found in Section 4). The unit-length assumption implies that the length of a path is the number of edges in the path, and enables us to normalize to 1 the transit time of each agent along an edge. The unit-capacity assumption says that at most one agent can leave the buffer (defined below) of an edge at each integer time point.

2.4. Buffer regulations. Throughout this paper (with the exception of Section 4), we make the following two weak assumptions (also referred to as buffer regulations), which are satisfied by most related models in the literature:

(A1) The tail part of each edge has an imaginary buffer with an infinite capacity and no physical size. It is the entrance of the edge through which agents get into the edge at integer time points. As soon as an agent reaches a vertex that is not his destination, he enters the buffer of the next edge in his route immediately without any delay.

(A2) At each integer time point, whenever a buffer is nonempty, exactly one agent *leaves* from the buffer and starts to move along the edge (that contains the buffer) and other agents (if any) wait in the buffer for one more unit of time.

Our ADR model is very similar to the packet routing model in [24]. Assumption (A1) states that waiting agents are located at the tail parts of edges, while in [24] they are located on vertices. The difference is inessential. Assumption (A2), a property called *greedy* in [24], implies in particular that at any integer time point an agent has to leave the buffer if he is the only one in it. Note that (A2) implies that waiting times imposed by any agent on other agents in one buffer are one unit of time. Note also that there is no specific rule in (A2) for determining the leaving agent from a buffer when there are multiple agents in it.

2.5. ADR dynamics. It takes two types of time for an agent to use an edge: a unit transit time from the tail vertex (buffer) to the head vertex, and a variable amount of waiting time in the buffer, which is determined by the above buffer regulations along with certain rule for selecting the agent to leave the buffer. Since agents spend no time on vertices, the residence times, the differences between the times they enter and exit network G, can be defined as follows.

DEFINITION 1. The *residence time* of an agent in network G is defined as the total transit time plus the total waiting time spent on all the edges (their buffers) in his OD path.

By the unit-transit-time assumption on single edges, the total transit time of an agent equals the number of edges (i.e., length) of his OD path. Although time is continuous when considering edge traverses, only integer time points matter as far as entering and leaving buffers as well as reaching vertices are concerned (due to integer entry times into the network, buffer regulations (A1), (A2) and unit transit time on single edges). The following ADR dynamics makes this more precise.

- When an agent enters network G at (integer) time r, we say that he *reaches* his origin at time r. As (A1) specifies, the agent *enters* the buffer on the starting edge of his OD path at the same time r.
- If an agent leaves a buffer on edge e (which is located at e's tail part) at (integer) time r, then it follows from (A2) and unit transit time on e that the agent moves along e during time interval $(r, r+1)$ until he *reaches* the head vertex of e at time $r+1$.
- When an agent reaches a vertex that is not his destination at (integer) time r, by (A1), he *enters* the buffer of the next edge in his route immediately at the same time r.
- When an agent reaches his destination at time r, we assume that he *exits* G immediately at the same time r.

Having presented the dynamics of our ADR model, we conclude this section by introducing some terms we will use to describe agents' status under ADR. Consider an arbitrary agent $i \in \Delta$, who enters G from origin o_i at time s and exits G from destination d_i at time t. We say that agent i is *inside* G during time period $[s, t)$. Since the action of leaving a buffer is instantaneous, there is no confusion to assume that when i leaves a buffer at time r, he is also in the buffer at time r. Hence for each integer point $r \in [s, t)$, agent i is in a unique buffer on his OD path. We say that agent i *delays* agent j on edge e (at time r) if at that time i leaves and j stays in e's buffer. By (A2), an agent starts waiting for one more unit of time whenever he is delayed by someone else. Therefore, the total waiting time of each agent is also the number of times he is delayed in all buffers he uses.

If two agents stay in the buffer of the same edge e at the same time, then one of them delays the other exactly once on e but not vise versa (recall that the network is acyclic).

3. Bounding residence times in unit-capacity networks.

In this section, we are given a routing (an ADR profile) $\boldsymbol{p}$ on the unit-capacity network G. We focus on an arbitrarily fixed agent $\zeta \in \Delta$ and reserve the symbol r for ζ's entry time into G. Let G_ζ be the subnetwork of G that is composed of all o_ζ-d_ζ paths in G.

3.1. Main results. Let r_0 $(\geq r)$ denote any fixed integer time point at which agent ζ is still inside the network G. Two sets of agents, $A_{\zeta,\boldsymbol{p}}$ and $B_{\zeta,\boldsymbol{p}}(r_0)$ defined below, are crucial in bounding ζ's residence time:

- $A_{\zeta,\boldsymbol{p}}$ is the set of agents who, under $\boldsymbol{p}$, (i) enter G later than agent ζ (i.e., after time r) and (ii) use at least one edge of G_ζ.
- $B_{\zeta,\boldsymbol{p}}(r_0)$ is the set of agents other than ζ who, under $\boldsymbol{p}$, (i) enter G no later than agent ζ, (ii) are still inside G at time r_0, and (iii) use at least one edge of G_ζ.

Note that $A_{\zeta,\boldsymbol{p}}$ and $B_{\zeta,\boldsymbol{p}}(r_0)$ are disjoint, agent ζ belongs to neither $A_{\zeta,\boldsymbol{p}}$ nor $B_{\zeta,\boldsymbol{p}}(r_0)$, and there may be agents in $\Delta\backslash\{\zeta\}$ who are neither in $A_{\zeta,\boldsymbol{p}}$ nor in $B_{\zeta,\boldsymbol{p}}(r_0)$.

For any agent $i \in A_{\zeta,\boldsymbol{p}} \cup B_{\zeta,\boldsymbol{p}}(r_0) \cup \{\zeta\}$, since G is acyclic, the intersection of i's route and G_ζ must be a *nontrivial* path (i.e., a path of at least one edge). When i reaches the last vertex (i.e., sink vertex) of that path (the acyclic property implies that he has finished his journey on the path and therefore that in G_ζ), we say that he *exits* G_ζ. For any vertices $v, w \in V$, let L_{vw} denote the length of a longest v-w path in G. In particular, $L_{o_\zeta d_\zeta}$ is the length of a longest path that ζ may traverse in the network.

THEOREM 1. *Given an ADR profile $\boldsymbol{p}$ on a unit-capacity network G, for any integer time point $r_0 \geq r$, if no agent in $A_{\zeta,\boldsymbol{p}}$ exits G_ζ before agent ζ, then ζ exits the network no later than time $r_0 + L_{o_\zeta d_\zeta} + |B_{\zeta,\boldsymbol{p}}(r_0)|$.*

The proof of Theorem 1 will be presented later in this section. Theorem 1 provides a series of possible upper bounds of ζ's residence time parameterized by r_0, some of which will be used in Section 6 to handle the EQLP. It is easy to see the condition that "no agent in $A_{\zeta,\boldsymbol{p}}$ exits G_ζ before ζ" is not redundant, because otherwise the focal agent ζ may be delayed by an infinite number of later coming agents (e.g., the buffers always give priorities to other agents for leaving), which makes no upper bound possible. This condition is easily met in many situations (e.g., those to be introduced in Section 5). In particular, the condition is always valid when $A_{\zeta,\boldsymbol{p}} = \emptyset$. We state the corresponding result as a formal corollary as follows.

COROLLARY 1. *Given an ADR profile $\boldsymbol{p}$ on a unit-capacity network G, if $A_{\zeta,\boldsymbol{p}} = \emptyset$, then the total residence time of agent ζ in the network is at most $L_{o_\zeta d_\zeta} + |B_{\zeta,\boldsymbol{p}}(r)|$.*

A case even more special than $A_{\zeta,\boldsymbol{p}} = \emptyset$ is that all agents are in the network at the very beginning, for which we additionally have $r = 0$ and $|B_{\zeta,\boldsymbol{p}}(0)| = |\Delta| - 1$. We explain in the following that even for this special case, Corollary 1 is nontrivial.

An immediate intuition is that agent ζ can be delayed by any other specific agent at most once. Since $L_{o_\zeta d_\zeta}$ is the longest possible transit time that ζ spends and ζ could be delayed by at most $|\Delta| - 1$ other agents, the upper bound $L_{o_\zeta d_\zeta} + |\Delta| - 1$ seems trivial. However, this intuition is incorrect even for the simplest case of $|\Delta| = 2$. In fact, ζ may be delayed by other agents, even one particular agent, for more than $|\Delta|$ times. This indicates that the upper bound is an integrated one that cannot be easily decomposed. The example below shows that this upper bound is tight even if the network has only one OD pair.

EXAMPLE 1 (TIGHTNESS). The network G with a single OD pair (o,d) is as depicted in Figure 1, with the length $L \equiv L_{od}$ of the longest o-d paths being 9. All $|\Delta| = 3$ agents, indexed 1, 2 and 3, enter G at the same time from their common origin o. Agent $i \in \{1,2,3\}$ follows the unique o-d path that uses edges e, e_i, e'_i. The routing obeys the agent-priority rule that an agent can only be delayed by lower indexed agents. It can be verified that the residence time of agent 3 is 11, which is precisely $L + |\Delta| - 1$. Note also that this upper bound is tight in the following stronger sense: even if agent 3 knows the choices of agents 1 and 2 and best responds to them, his residence time will still reach the same upper bound.

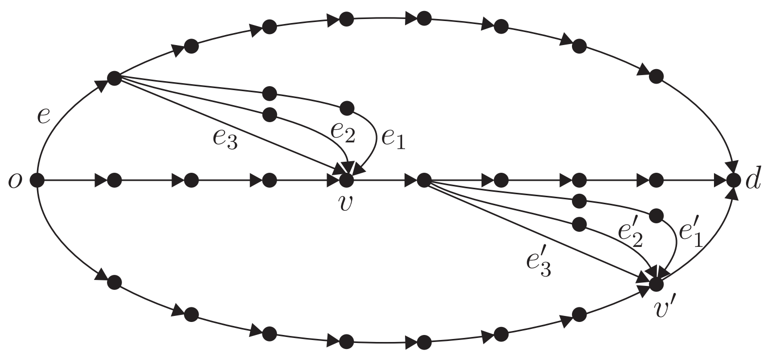

FIGURE 1. An example for the tightness of the upper bound $L + |\Delta| - 1$

REMARK 1. The agent-priority rule adopted in Example 1 is the one investigated in [12] (referred to as (R1) in Section 5 of this paper). Example 1 can also be easily modified to show the tightness for buffer selection rules studied in [28] and [3] (referred to as (R2) and (R3), respectively, in Section 5 of this paper).

REMARK 2. The acyclicness of the input network is necessary for the upper bound to be $L + |\Delta| - 1$: If the underlying network contains a cycle, Cidon et al. [5] construct an example with a lower bound of $\Omega(\sqrt{|\Delta|}L_{o_\zeta d_\zeta} + |\Delta|)$ on the residence time of agent ζ.

3.2. Technical details. In the remainder of this section, we explore a multi-token technique to prove Theorem 1.

Token production and movement. For any agent i and integer time point $s \geq r_0$ such that i is inside G during time interval $[s, s+1)$, let $U_{i,s}$ denote the buffer where i stays at time s. Clearly $U_{i,s}$ is well defined. Agent ζ will produce tokens over time with indices $1, 2, \ldots$ inside the buffers he stays in, at most one token being produced at a time and each production taking no time. The tokens will move along with the agents who hold them currently. The precise procedure of token productions (which happen inside buffers $U_{\zeta,r_0}, \ldots$) and token movements (which happen along with agents' movements) is as follows:

- Initially (at time r_0), agent ζ produces a token with index 1 inside buffer U_{ζ,r_0}; no other tokens are inside G.
- Iteratively, for each time point $s = r_0, r_0 + 1, \ldots,$

(s1) If the buffer $U_{\zeta,s}$ contains no token at time s, then ζ produces, at time s, a new token inside $U_{\zeta,s}$ with an index being the largest existing token index plus one;

(s2) For every agent i who leaves a buffer $U_{i,s}$ at time s, if $U_{i,s}$ contains token(s), then the one with the smallest index among them is *held* by and *moves* along with agent i during time interval $(s, s+1)$; as a result of the movement, at time $s+1$, the token and agent i either are in the same buffer $U_{i,s+1}$ or exit G from d_i together.

Note that the productions and movements of tokens do not affect the original dynamics in ADR.

Properties. The following observations are helpful for our later discussions.

(p1) At any time point, each agent holds at most one token, and the number of tokens inside a buffer is no more than that of the agents in the same buffer;

(p2) At any time point s for which $U_{\zeta,s}$ is well defined, the agent who leaves $U_{\zeta,s}$ holds exactly one token; in particular, agent ζ holds a token whenever he moves;

(p3) At any time point $s > r_0$, agent ζ produces a new token (with index at least 2) in buffer $U_{\zeta,s}$ only if he stays in the same buffer during $[s-1, s]$.

Apparently, the trajectory along which each token moves is a path in G that starts from the edge where the token is produced and ends at some destination d_i. More precisely, the *motion trail* of each token (with index) k, denoted by T_k, is an o_ζ-d_i path or a proper sub-path of an o_ζ-d_i path for some $i \in \Delta$. For any token k, since G is acyclic, the intersection of T_k and G_ζ is a nontrivial path, which we write as $T_k \cap G_\zeta$. Let v be a vertex on $T_k \cap G_\zeta$. We say that token k *departs from* v if one of the following conditions is satisfied: (i) v is not the sink vertex of path $T_k \cap G_\zeta$ and k starts to move along $\epsilon_v^+(T_k \cap G_\zeta)$, i.e., the edge on $T_k \cap G_\zeta$ outgoing from v; (ii) v is the sink vertex of $T_k \cap G_\zeta$ and k reaches v. In the case (ii), we also say that token k exits G_ζ, because at that time the agent holding k exits G_ζ (recall the definition above Theorem 1).

The following lemma implies directly that each token k exits G_ζ no later than time $r_0 + (L_{o_\zeta d_\zeta} + k - 1)$.

Lemma 1. *For every vertex $v \in T_k \cap G_\zeta$, token k departs from v no later than $r_0 + (L_{o_\zeta d_\zeta} - L_{v d_\zeta} + k - 1)$.*

Proof. Our proof is by induction on token index k. For the base case of $k = 1$, we need to show that token 1 departs from vertex v no later than $r_0 + (L_{o_\zeta d_\zeta} - L_{v d_\zeta})$. Observe that token 1 is the "smallest" and never waits during the whole moving process. Therefore, the total time token 1 spends in G_ζ before departing from v is the length of a subpath of an o_ζ-v path (recall that the first vertex token 1 departs from is the tail vertex of the edge where ζ stays at time r_0), which is at most $L_{o_\zeta v} \le L_{o_\zeta d_\zeta} - L_{v d_\zeta}$, where the inequality is guaranteed by the fact that G is acyclic.

Suppose now $k \ge 2$ and Lemma 1 holds for tokens $1, \ldots, k-1$. For token k, consider first the easier cases that k departs from v right after some token h $(< k)$ through edge $\epsilon_v^+(T_k \cap G_\zeta)$. Let V_k denote the set of these vertices v.

Claim 1. *For every $v \in V_k$, token k departs from v no later than $r_0 + (L_{o_\zeta d_\zeta} - L_{v d_\zeta} + k - 1)$.*

By the induction hypothesis, token h departs from vertex v no later than $r_0 + (L_{o_\zeta d_\zeta} - L_{v d_\zeta} + h - 1)$. It follows that token k departs from v no later than $r_0 + (L_{o_\zeta d_\zeta} - L_{v d_\zeta} + h) \le r_0 + (L_{o_\zeta d_\zeta} - L_{v d_\zeta} + k - 1)$, proving the claim.

We are left to consider the remaining case of vertex $v \in (T_k \cap G_\zeta) \backslash V_k$, which particularly implies that token k does not wait on edge $\epsilon_v^+(T_k \cap G_\zeta)$. Note that just one time unit before token k (≥ 2) is produced, agent ζ is waiting on the starting edge of T_k for some other agent to leave the buffer (according to (p3)), and the agent holds a token (according to (p2)) whose index is smaller than k (according to (s1)). It follows that the starting vertex of T_k belongs to V_k. Let $v' \in T_k \cap V_k$ be the vertex nearest to v such that

- Token k departs from vertex v' before it departs from v;
- Vertex v' is the unique vertex in the path $T_k[v', v]$ that belongs to V_k.

The choice of v' and (s2) enforce that token k never waits from the time it departs from v' to the time it departs from v. According to Claim 1, token k departs from v' no later than $r_0 + (L_{o_\zeta d_\zeta} - L_{v' d_\zeta} + k - 1)$. Since the total time token k spends on the non-stop movement from v' to v is at most $L_{v'v}$, token k departs from v no later than $r_0 + (L_{o_\zeta d_\zeta} - L_{v' d_\zeta} + k - 1) + L_{v'v} \le r_0 + (L_{o_\zeta d_\zeta} - L_{v d_\zeta} + k - 1)$, where $L_{v' d_\zeta} \ge L_{v'v} + L_{v d_\zeta}$ is implied by the fact that G_ζ is acyclic. This proves Lemma 1. □

Now we are ready to prove the main result of this section.

Proof of Theorem 1. If agent ζ holds a token no larger than $|B_{\zeta,\boldsymbol{p}}(r_0)|+1$ when he reaches d_ζ, then by Lemma 1 we have already obtained the desired result.

Suppose the index of the token that ζ holds when reaching the destination d_ζ is at least $|B_{\zeta,\boldsymbol{p}}(r_0)|+2$. By Lemma 1, every token k with $1\le k\le |B_{\zeta,\boldsymbol{p}}(r_0)|+1$ exits G_ζ no later than $r_0+(L_{o_\zeta d_\zeta}+k-1)\le r_0+(L_{o_\zeta d_\zeta}+|B_{\zeta,\boldsymbol{p}}(r_0)|)$, along with some agent $i(k)$ ($\neq\zeta$) whose OD path under $\boldsymbol{p}$ uses at least one edge of G_ζ. Recall from (p1) that each agent can hold at most one token when exiting G_ζ. We claim that there is at least one $k\in[1,|B_{\zeta,\boldsymbol{p}}(r_0)|+1]$ with $i(k)\notin B_{\zeta,\boldsymbol{p}}(r_0)$. The claim is true because none of the $|B_{\zeta,\boldsymbol{p}}(r_0)|+1$ agents $i(1),\dots,i(|B_{\zeta,\boldsymbol{p}}(r_0)|+1)$ is ζ, and they cannot all belong to $B_{\zeta,\boldsymbol{p}}(r_0)$. It follows from $i(k)\notin B_{\zeta,\boldsymbol{p}}(r_0)$ that $i(k)$ enters G later than agent ζ, and thus belongs to $A_{\zeta,\boldsymbol{p}}$. Now we are done, because the hypothesis of the theorem says that no agent in $A_{\zeta,\boldsymbol{p}}$ exits G_ζ before agent ζ, from which we deduce that ζ reaches d_ζ no later than the time agent $i(k)$ exits G_ζ and hence no later than $r_0+(L_{o_\zeta d_\zeta}+|B_{\zeta,\boldsymbol{p}}(r_0)|)$. □

4. Bounding residence times in uniform-capacity networks.

A natural question is whether Theorem 1 can be generalized to any network with arbitrary integer edge-capacities and arbitrary integer edge-lengths. The answer is yes, because subdividing and splitting the edges into the ones with unit capacity and unit length and letting agents move in a corresponding way does not change agents' residence times. However, this answer is not completely satisfactory because when roads are wider, one may expect shorter residence times. In this section, we consider a clean scenario where all roads have the same integer capacity of c. So a more meaningful question is, for this scenario, do we have results parameterized with c that reduce to Theorem 1 when $c=1$?

Let us first elaborate on this more general model. Given any positive integer c, let G_c be the network with the same vertex and edge sets as G in which each edge has a capacity of c, meaning that at most c agents can move along an edge at the same time. The buffer regulations, when applied to G_c, keep condition (A1) unchanged, and relax (A2) by allowing exactly $n_s(U):=\min\{c,n\}$ agents in a buffer U on edge e to leave U at the same time point s, where n denotes the number of agents in U at time s; these $n_s(U)$ agents reach e's head vertex at the next time point $s+1$ simultaneously.

It turns out that obtaining a generalized result mentioned in the above question is nontrivial. In this section, we explore a single-token technique to provide a satisfactory answer. As a trade-off, we have to make an additional assumption: all (finitely many) agents enter the network at the very beginning. As in the preceding section, we focus on an arbitrarily fixed agent $\zeta\in\Delta$ throughout this section.

Besides the difference in the numbers of tokens (as their names indicate), the single and multi-token techniques differ from each other mainly in two aspects. First, while the set of tokens is not fixed but produced dynamically in the multi-token analysis, the unique token is available at the beginning of our single-token analysis. Second, while the moving of tokens is completely determined by the buffer regulations in the multi-token analysis, the moving of the unique token is relatively free in the single-token analysis, as detailed in the following.

(c0) Initially (at time 0), the token is produced inside the buffer where agent ζ stays.

The token will move along edges together with some agent, and finally reach the destination d_ζ at the same time as agent ζ. During the process, the token may wait for some time in buffers. Specifically, the token's moving and waiting must be accompanied by some agent, for which the following conditions must be satisfied:

(c1) When the token moves along an edge e to the next buffer during time period $(s,s+1)$, there must be an agent moving along e to the same buffer during $(s,s+1)$; we may think of the token as being held by the agent during $(s,s+1)$ and when restricted to e, the movements of the token and the agent are identical. When the token waits in a buffer during time period $(s,s+1)$, there must be an agent staying in the same buffer during $(s,s+1)$.

(c2) The token reaches its destination d_ζ (and exits G) at the same time as agent ζ. (At that time the agent holding the token might not be ζ.)

Under the above conditions, we may assume that at any time point, the token is held by exactly one agent. Conditions (c1) and (c2) are obviously satisfied if the token is always held by agent ζ.

Definition 2. We call every moving dynamic of the single token that satisfies conditions (c0)–(c2) a *legal dynamic*.

Let f denote the starting edge of agent ζ's route. Given any legal dynamic, since G is acyclic, it is clear that the token's motion trail under the dynamic is an o_ζ-d_ζ path P with starting edge f. We write the path as the concatenation of its edges in sequence $P=e_1e_2\cdots e_k$ with edge $e_1=f$ and the head of edge e_k being d_ζ. The legal dynamic is then represented by $D=(P;t(e_1),t(e_2),\dots,t(e_k))$, where for each $i=1,\dots,k$, $t(e_i)$ is the time when the token leaves e_i's buffer under the dynamic. Clearly, $t(e_1)<t(e_2)<\cdots<t(e_k)$. Suppose that ζ reaches d_ζ at time τ_ζ. Then (c2) along with the unit transit time on e_k implies $t(e_k)=\tau_\zeta-1$.

Let $\mathcal{D}$ be the set of all legal dynamics, which is nonempty because the special dynamic in which the token is always held by ζ is legal. In the following, we define a partial order among all legal dynamics in $\mathcal{D}$.

Definition 3 (Order on token dynamics). Dynamic $(P;t(e_1),t(e_2),\dots,t(e_k))\in\mathcal{D}$ is said to be (strictly) *smaller* than dynamic $(P';t'(e'_1),t'(e'_2),\dots,t'(e'_{k'}))\in\mathcal{D}$ if $t(e_1)<t'(e'_1)$ (note $e_1=e'_1=f$), or there exists $i\ge 2$ such that $e_h=e'_h$ and $t(e_h)=t'(e'_h)$ for all $h=1,\dots,i-1$, $e_i=e'_i$ and $t(e_i)<t(e'_i)$.

Claim 2. *The binary relation on $\mathcal{D}$ defined in Definition 3 gives rise to a partial order on $\mathcal{D}$, i.e., it is anti-symmetric and transitive.*

The proof is a simple exercise, which is relegated to Appendix A. It follows from Claim 2 that a minimal legal dynamic exists. Intuitively, a minimal legal dynamic is such that the token leaves every buffer as soon as possible, so long as it can exit the network at the same time as ζ.

In the token's moving (and waiting) process, we say that the token is *delayed* by an agent i on edge e, if there exists integer $s\ge 0$ such that during $(s,s+1)$ the token waits (stays) in the buffer on e and agent i moves along e.

Theorem 2. *Given any ADR profile on G_c, suppose all the $|\Delta|$ agents enter G_c at the same time. Then the residence time of agent ζ is at most $L_{o_\zeta d_\zeta}+\lfloor(|\Delta|-1)/c\rfloor$.*

Proof. Since by (c2) the residence time of the token in any legal dynamic is the same as that of ζ, it suffices to show that there exists one legal dynamic under which the token's residence time is at most $L_{o_\zeta d_\zeta}+\lfloor(|\Delta|-1)/c\rfloor$. As it takes one unit of time for the token to move (transit) along any edge and the length of the longest o_ζ-d_ζ path is $L_{o_\zeta d_\zeta}$, the token's total transit time under any legal dynamic is upper bounded by $L_{o_\zeta d_\zeta}$. Therefore, it suffices to show that there exists one legal dynamic under which the token's total waiting time is at most $\lfloor(|\Delta|-1)/c\rfloor$. We prove that any minimal legal dynamic $D^*=(P^*;t(e^*_1),t(e^*_2),\dots,t(e^*_k))$, whose existence is guaranteed by Claim 2, meets the requirement. Observe that the buffer regulations on G_c imply that whenever the token is delayed by an agent, it (together with the agent holding it) is delayed by precisely c agents simultaneously. Therefore, it suffices to show that under D^*, the token is never delayed by ζ and is delayed by any other specific agent at most once.

First, the token is never delayed by ζ. Otherwise, we can modify the dynamic in such a way that, ever since the first time ζ is about to delay the token, he holds the token all the remaining way to the destination d_ζ. The new dynamic is still legal but smaller than D^*, a contradiction. Now suppose the token is delayed by an agent η twice on edge e^*_i and e^*_j, respectively, where

$1 \le i < j \le k$. This means that when agent η leaves e_i^*'s (resp. e_j^*'s) buffer, the token under D^* is over there; the token will leave the buffer at a later time. Let $e_i^* e_1' \cdots e_\ell' e_j^*$ be the sub-path of η's route from e_i^* to e_j^* and $\tilde{t}(e_i^*), \tilde{t}(e_1'), \ldots, \tilde{t}(e_\ell'), \tilde{t}(e_j^*)$ be the time points when η leaves the buffers of $e_i^*, e_1', \ldots, e_\ell', e_j^*$, respectively. Then we have $\tilde{t}(e_i^*) < t(e_i^*)$.

Let $P' := e_1^* \cdots e_i^* e_1' \cdots e_\ell' e_j^* \cdots e_k^*$ be obtained from P^* by replacing its sub-path from e_i^* to e_j^* with $e_i^* e_1' \cdots e_\ell' e_j^*$. Since G is acyclic, P' is an o_ζ-d_ζ path. We modify dynamic D^* to $D' = (P'; t(e_1^*), \ldots, t(e_{i-1}^*), \tilde{t}(e_i^*), \tilde{t}(e_1'), \ldots, \tilde{t}(e_\ell'), t(e_j^*), \ldots, t(e_k^*))$ by letting agent η hold the token when he moves along the sub-path $e_i^* e_1' \cdots e_\ell'$. Since under D^* the token is in e_i^*'s (resp. e_j^*'s) buffer at time $\tilde{t}(e_i^*)$ (resp. $\tilde{t}(e_j^*)$), we see that $D' \in \mathcal{D}$ is a legal dynamic. However, $\tilde{t}(e_i^*) < t(e_i^*)$ shows that D' is smaller than D^*, contradicting the choice of D^*. □

As a byproduct, this new technique also enables us to obtain a simpler proof of the main result by Mansour and Pattshamir on shortest path routing [24]. See Proposition 1 in Appendix B for details. In fact, Theorem 2 can be considered as a generalization of the result from shortest path networks to more general ones, which is listed in [24] as an open problem.

5. Games of atomic dynamic routing.

By providing preliminaries for three ADR games, this section serves as a bridge between the preceding sections and the next one. We focus on the ADR games in acyclic directed network $G = (V, E)$ with a single OD pair (o, d). If not otherwise specified, we always assume G is unit-capacitated. For each integer $r \ge 0$, the set of agents who enter G at time r is denoted Δ_r, which is assumed to be of a finite size.

Theorem 1 tells us that the residence time of an arbitrary agent ζ entering network G at time r is upper bounded by L_{od} plus some number $\beta(r)$, provided that later coming agents do not exit G earlier than ζ. This bound is not completely satisfactory, because while L_{od} is a constant determined by the input network G, the number $\beta(r)$ may grow with time r and hence is generally not a constant. A natural question arises: is it possible to bound all $\beta(r)$ with a constant that is determined by the input network? The answer is obviously “No”, because continuously incoming agents may be congested gradually on some edge and thus $\beta(r)$ may grow to infinity.

When we consider selfish and rational agents (for which the routing model is often considered a noncooperative game), however, the above question becomes more interesting, because when an edge is very congested, intuitively later coming agents will avoid using it, making it less congested. In order to obtain a constant bound on $\beta(r)$, an additional necessary requirement is clear: the (average) inflow size should not exceed the network capacity. The question on the boundedness of $\beta(r)$ is closely related to the EQLP introduced in Section 1: are the lengths of queues at equilibrium bounded? The answer to the EQLP is “Yes” if and only if all $\beta(r)$ under the equilibrium routing are bounded. It turns out that the results we obtain in Section 3 for bounding residence times in ADR are useful to our resolution of the EQLP on series-parallel networks.

5.1. Three game models. Building on the general ADR model introduced in Section 2, we specify below three additional buffer regulation rules for selecting the leaving agent from a buffer. The addition of each of the rules makes the arrival time of an agent at a vertex under any given path profile uniquely determined, which leads to a well-defined (deterministic) ADR game Γ_i, $i \in \{1, 2, 3\}$.

Buffer selection rules. In defining the agent selection rules for games Γ_1 and Γ_2, we are given a *global agent-priority list*, i.e., a complete ordering among all agents, which assumes that an agent entering G earlier always has a higher priority over any agent entering G later. It follows from this assumption that agent priority is a refinement of the relative order of their entry times.

(R1) In ADR game Γ_1, among all agents concurrently staying in a buffer, the agent with the highest priority leaves the buffer first.

On the other hand, the selection rules of games Γ_2 and Γ_3 both follow the first-in-first-out (FIFO) principle — the agent who enters a buffer earlier has a higher FIFO rank and leaves the buffer earlier. (Note that game Γ_1 does not necessarily satisfy this property.) However, games Γ_2 and Γ_3 use different priority rules to break a tie when multiple agents enter the buffer at the same time. A readily made tie-breaking rule resorts to the given global agent-priority list:

(R2) In ADR game Γ_2, in determining which agent to leave a buffer first in the case of a tie in following the FIFO principle, the agent with the highest priority is selected.

Another method for breaking a FIFO tie is to use edge priority, where each vertex $v \in V\backslash\{o\}$ is associated with a fixed priority ordering among all incoming edges to v. Besides, for each set Δ_r of agents who enter G at the same time r, we are given an *original agent-priority list* of these agents (i.e., a complete ordering among them), to break ties only in the first buffers they enter.

(R3) In ADR game Γ_3, in determining which agent to leave a buffer first in the case of a tie in following the FIFO principle, it is resolved by the following rules applicable to two different scenarios:

- Edge priorities: if two agents enter the buffer simultaneously from different incoming edges, then the agent coming from the edge with a higher priority is selected earlier.
- Original agent-priority lists: if two agents enter the buffer simultaneously when they enter G, then the agent with a higher original priority is selected earlier.

When restricted to acyclic networks and non-zero edge lengths (as is assumed in our paper), the store-and-forward packet routing studied in [12] is actually a special case of game Γ_1, where all (finitely many) agents are in the network at the very beginning. Games Γ_2 and Γ_3 have been studied in [28] and [3], respectively. (An edge-priority rule similar to Γ_3 has been discussed in [33].) In contrast to the global agent-priority list in Γ_1 and Γ_2, which orders all agents and works for all buffers, the original agent-priority lists in Γ_3 are local in that each of them orders only a subset of the agents and works only at the origin for their first buffers.

Equilibria. Recall that $\Delta = \cup_{r\geq 0}\Delta_r$ denotes the agent set. Let $\mathscr{P}$ denote the set of OD paths in network G. For any path profile $\boldsymbol{p} = (P_i)_{i\in\Delta}$ (with $P_i \in \mathscr{P}$ for all $i \in \Delta$) of the game, any agent $j \in \Delta$, and any vertex $v \in V$, we use $t_j^v(\boldsymbol{p})$ to denote the arrival time of agent j at v under routing $\boldsymbol{p}$ (in particular, $t_j^v(\boldsymbol{p}) = \infty$ if $v \notin P_j$). All agents $j \in \Delta$ are selfish, trying to choose a path P_j to minimize $t_j^d(\boldsymbol{p})$.

All selfish agents make their routing decisions only at the very start of the game (i.e., at time 0) as to which OD path to take, no matter what times they enter the network G. In game-theoretic terminology, this is a game of normal form (or a simultaneous-move game), whose standard solution concept is the *Nash equilibrium* (NE).

Definition 4 (NE). A path profile $\boldsymbol{p}$ of Δ is an *NE* of the ADR game if no agent can gain by a uniliteral deviation, i.e., $t_i^d(\boldsymbol{p}) \leq t_i^d(P_i', \boldsymbol{p}_{-i})$ for all $i \in \Delta$ and $P_i' \in \mathscr{P}$, where $\boldsymbol{p}_{-i}$ is the partial path profile of $\boldsymbol{p}$ for agents in $\Delta\backslash\{i\}$.

The following concept of *uniformly fastest route* (UFR) *equilibrium* plays a central role in [28]. In ADR games Γ_1 and Γ_2, UFR equilibria form a nonempty proper subset of the NE set.

Definition 5 (UFR equilibrium). A path profile $\boldsymbol{p}$ is a *UFR equilibrium* if for every agent $i \in \Delta$ and every vertex v on i's OD path under $\boldsymbol{p}$, there is no $P_i' \in \mathscr{P}$ such that $t_i^v(P_i', \boldsymbol{p}_{-i}) < t_i^v(\boldsymbol{p})$.

For game Γ_3, we shall study general NEs, under which agents are well-behaved [3] (see Theorem 3 below). For games Γ_1 and Γ_2, however, we shall focus on UFR equilibria, because agents under a general NE in these two models are not so well-behaved, and no effective method has been found to analyze their general NEs.

5.2. Properties of equilibria. We show in this subsection that the equilibria in the three ADR game models Γ_1, Γ_2 and Γ_3 are quite regular, paving the way for our later analysis. To make our exposition more precise, we make the following definition.

Definition 6 (Preemption). Given a routing for one of the three ADR games Γ_1, Γ_2 and Γ_3, we say that agent i *preempts* agent j at vertex v if (i) they both pass v, and agent i reaches v earlier than agent j, or (ii) i and j reach v at the same time but the following holds:

- For Γ_1 and Γ_2, agent i has a higher priority than agent j;
- For Γ_3, agent i comes from an edge in E_v^- with a higher priority than agent j when $v \neq o$, and i has a higher (original) priority than j when $v = o$.

Note that when agent i preempts agent j, they may not meet on any edge at all, and hence it is possible that no one delays the other. However, when i delays j, it must be true that i preempts j at the head vertex of the corresponding edge.

In the cases of Γ_1 and Γ_2, suppose that agent i preempts agent j at origin o. This is equivalent to that i has a higher priority than j (recalling from the assumption on the global agent-priority list that lower-priority agents enter G no earlier than higher-priority ones). The following property that j can never preempt i admits an algorithmic proof, which is deferred to Appendix C (see Lemma 8(i) over there).

Lemma 2. *Under every UFR equilibrium of Γ_1 or Γ_2, no agent with a lower priority can preempt an agent with a higher priority.*

Given a routing, agents who reach the common destination d (and exit G) at the same time form a *batch*. As seen from the following theorem, at equilibrium of the three game models, agents of different batches move through the network in an hieratically independent way.

Theorem 3. *Consider any of the ADR games, Γ_1, Γ_2 or Γ_3, on unit-capacity network G with a single OD pair (o, d). Let $\mathbf{p}$ be a UFR equilibrium if the game is Γ_1 or Γ_2, and an NE if the game is Γ_3. The following properties are satisfied.*

(i) Global FIFO: *Under equilibrium $\mathbf{p}$, if agent i preempts agent j at some vertex (including the origin o), then i reaches the destination d no later than j.*

(ii) No Overtaking: *If agents of a batch and those in all earlier batches follow their routes as in equilibrium $\mathbf{p}$, then no matter which routes the later-batch agents choose, the former agents preempt all latter agents at every common vertex of their routes (i.e., the former agents cannot be overtaken by the latter ones). Therefore, the arrival times of the former agents at all vertices of their routes cannot be influenced by the latter agents.*

In the cases of Γ_1 and Γ_2, Theorem 3 follows from Lemma 8 in Appendix C. In the case of Γ_3, Theorem 3(i) is from Theorem EC.6 in [4], and Theorem 3(ii) is a direct consequence of Theorem 2 in [4].

Theorem 3 is essential for us to deal with the EQLP in the next section. It also implies that the condition in Theorem 1 – agents entering the network later than agent ζ do not exit the network earlier than him – is not too restrictive. Furthermore, since the three relevant models are very representative in the recent literature on ADR games and the properties are natural and intuitive, Theorem 3 is also of interest in its own right, rather than merely technical.

6. Bounding equilibrium queue lengths.

This section gives an affirmative answer to the EQLP on series-parallel networks for the three ADR games introduced in the preceding section. Below is one of the most popular definitions of this class of networks, expressed by iterative construction.

Definition 7 (Series-parallel network). A network G with origin o and destination d is *o-d series-parallel* or simply *series-parallel* if one of the following is true:

(i) G consists of a single edge od.

(ii) G is obtained by connecting two smaller o_i-d_i series-parallel networks G_i, $i=1,2$, *in series* — merge d_1 and o_2, and rename o_1 as o and d_2 as d.

(iii) G is obtained by connecting two smaller o_i-d_i series-parallel networks G_i, $i=1,2$, *in parallel* — merge o_1 and o_2 into o and merge d_1 and d_2 into d.

Let $G=(V,E)$ be a series-parallel network. In view of Definition 7, network G is obtained from $|E|$ edges by performing a sequence of $|E|-1$ series or parallel connection operations. Each of these operations connects two series-parallel subnetworks G_1 and G_2 of G into a bigger series-parallel subnetwork G_3 of G. Fix any such sequence of $|E|-1$ operations that leads to G, and let $\mathfrak{S}$ be the set of all the subnetworks G_1, G_2 and G_3 that appear during the whole process of the sequence of $|E|-1$ connection operations. Then clearly,

$$G\in\mathfrak{S} \text{ and } |\mathfrak{S}|=2|E|-1.$$

Equilibrium. In the remainder of this section, we use $\boldsymbol{p}=(P_i)_{i\in\Delta}$ to denote an arbitrary UFR equilibrium if the underlying game is Γ_1 or Γ_2 and an arbitrary NE in the case of Γ_3. For brevity, we simply call $\boldsymbol{p}$ an *equilibrium* of the game on G.

EQLP. By an *o-d cut*, or simply a *cut*, of G, we mean a set of edges whose removal from G leaves the graph unconnected from o to d. Let the capacity of G be defined as the minimum number of edges in an o-d cut in G. The EQLP on the ADR game can be rephrased as: Can the number of agents inside G under $\boldsymbol{p}$ be bounded by a constant, assuming that the inflow size $|\Delta_r|$ is at most the capacity of G for all time points r?

To tackle the EQLP on G, the above inductive graphic structure might suggest a very natural approach — assuming the EQLP has an affirmative answer on both smaller series-parallel networks whose series or parallel combination is G, one tries to prove that the queue lengths under $\boldsymbol{p}$ are bounded. Unfortunately, this natural idea does not work in the case of parallel combination, because the inflow size of one of the smaller networks could exceed its capacity (although the inflow size under $\boldsymbol{p}$ is at most the capacity of G). To get around the difficulty caused by the unfavorable flow divisions between smaller networks in parallel, we come up with a novel induction base (see Definition 8 in Section 6.2), which is stronger than an affirmative answer to the EQLP. The success of our inductive method with this stronger base relies on an upper bound on the ratio between the populations in any pair of sub-networks in $\mathfrak{S}$ connected in parallel (as detailed in Section 6.1). A key to deriving such an upper bound is Theorem 1.

6.1. Population ratio.

Lemma 3 below states that the numbers of agents in the two subnetworks that are connected in parallel are in some sense balanced: it is impossible that one subnetwork is much more populated than the other. This is intuitive, because parallel subnetworks are substitutes for agents: they use at most one of them. The more populated a subnetwork is, the more congested it is. As one subnetwork becomes more congested compared with the other, its attractiveness decreases, and later (selfish) agents will have more incentives to use the other subnetwork, which will eventually make the two subnetworks relatively balanced in population. However, quantifying this simple intuition rigorously is nontrivial.

Let $D=\max_{v\in V}|E_v^-|$ denote the maximum in-degree of vertices in G, and L denote the length of a longest o-d path in G. At any integer time point, we say that a network *accommodates* an agent if the agent is inside some buffer in the network. Recall that $\boldsymbol{p}$ denotes an equilibrium of Γ_i with $i\in\{1,2,3\}$ on G.

Lemma 3. *Suppose that $G_1,G_2\in\mathfrak{S}$ are connected in parallel to form a series-parallel network in $\mathfrak{S}$, and they accommodate n_1 and n_2 agents, respectively, at some time under $\boldsymbol{p}$. Then $n_1\le D(L+n_2+1)$.*

To prove the above lemma, an intuitive approach seems reasonable: if a drastic imbalance of populations in the two subnetworks emerges, some agent in the much more populated subnetwork would benefit by switching his path to the less populated one, contradicting the definition of the equilibrium. However, there are two major obstacles.

• First, the two subnetworks typically do not form the whole network; the common sink vertex (destination) of the two subnetworks may not be the final destination d of the agents. Hence they may not have incentives to reach this intermediary vertex at an earlier time by switching paths. Recall that while the (global) property of earliest-arrival is satisfied by equilibrium routings in Γ_1 and Γ_2, it is generally not the case in Γ_3. This makes a direct derivation of a contradiction (to the definition of the equilibrium) unable to work.

• Second, it is not easy to evaluate what would happen if one agent had switched from a subnetwork to the other. The difficulty lies in two aspects: (i) The switching typically leads to a non-equilibrium routing. While agents behave relatively predictably at equilibrium, it is not the case for a non-equilibrium routing in general. In particular, in contrast to many other game-theoretical models, agents' latencies are not explicitly expressed as functions of closed form but determined by complicated dynamic processes in our models. (ii) Switching should have occurred before the time we investigate (the "some time" in Lemma 3), and thus unavoidable backtracking makes the counter-factual analysis even more complicated.

Fortunately, Theorems 1 and 3 established in Sections 3 and 5 can help us overcome these two obstacles. While the above intuitive approach intends to make a comparison for the single deviator, we accomplish our task by comparing the deviator with some earlier-batch agent. Roughly, though agents may not reach every intermediary vertex as early as possible at equilibrium, they (as Theorem 3(ii) implies) cannot be overtaken by any later-batch agent. This provides a different way of deriving a contradiction: finding a later-batch agent who can preempt an earlier-batch agent by switching his path to the other subnetwork. Technically, this is to evaluate the performance of a later-batch agent after his deviation. Theorem 1 can relate his exiting time from the subnetwork with population sizes and the longest-path length for the routing that is not necessarily an equilibrium, even when backtracking is involved.

Proof sketch. The remainder of this subsection is devoted to the proof of Lemma 3, which goes roughly as follows. We use a contradiction argument and suppose from now on that $n_1 > D(L+n_2+1)$. Instead of directly using the definition of an equilibrium and showing the existence of an agent who has a better choice than that in $\boldsymbol{p}$, we shall show the existence of an agent ζ in G_1 who, by switching his path, can overtake some agent of an earlier batch in $\boldsymbol{p}$, contradicting Theorem 3(ii).

Let o' and d' denote the common origin and destination of G_1 and G_2, respectively. Suppose that the time stated in Lemma 3 is r_0. Let S_1 and S_2 denote the sets of agents G_1 and G_2 accommodate at time r_0, respectively. Hence $|S_1| = n_1$ and $|S_2| = n_2$. We define ζ as an agent who enters G_1 (via o') under $\boldsymbol{p}$ the latest among all agents in S_1. Let r denote his entry time into G_1. Apparently $r \le r_0$. If there are more than one such agents, we take ζ to be the one who is preempted by all other such agents at o'. It follows from the global FIFO property in Theorem 3(i) that ζ is one of the latest agents among those in S_1 to reach the final destination d.

We consider a unilateral deviation of ζ from G_1 to G_2. The most innovative part of our proof is to show that there is a special path in G_2 which makes his arrival time at d' upper bounded by $r_0 + L + |S_2|$ (see Claim 7 below). Meanwhile, Theorem 3(ii) guarantees that this unilateral deviation does not affect any of the arrival times, particularly the arrival times at d', of agents in earlier batches under $\boldsymbol{p}$. Since the indegree of d' in G_1 is bounded by $D-1$, at most $D-1$ agents in S_1 can reach d' simultaneously. This enables us to easily derive a lower bound of $r_0 + \frac{|S_1|-(D+1)}{D-1}$ on the latest arrival time at d' among agents in S_1 who are in earlier batches than ζ (Claim 3). Using our contradiction assumption $n_1 > D(L+n_2+1)$, the above upper bound (Claim 7) and lower bound (Claim 3) help us find an agent (denoted as η) in S_1 who can be overtaken by ζ at d', contradicting the hypothesis that $\boldsymbol{p}$ is an equilibrium.

Technical details. We rigorously define the agent η and show that he would be overtaken by ζ. Let S_1' be a subset of S_1 with $D+1$ agents who reach d' under $\boldsymbol{p}$ as late as possible. Suppose that η is one of the agents who reaches the final destination d (of the whole network) the earliest among those in S_1'. The claim below provides a lower bound on η's arrival time at the destination d' of the subnetwork under the equilibrium routing $\boldsymbol{p}$.

Claim 3. $t_\eta^{d'}(\boldsymbol{p}) \geq r_0 + \frac{|S_1|-(D+1)}{D-1}$.

Proof. Since the in-degree of a vertex in G is at most D, and d' has at least one incoming edge from G_2, we know that the number of incoming edges of d' from G_1 is at most $D-1$. Hence at most $D-1$ agents in S_1 can reach the destination d' of G_1 at the same time. It follows from the definition of S_1' that the earliest time an agent in S_1' can reach d' is not earlier than $r_0 + |S_1 \setminus S_1'|/(D-1) = r_0 + \frac{|S_1|-(D+1)}{D-1}$ by definition of S_1'. As $\eta \in S_1'$, we have the claim. ☐

The following claim states that, as desired, agent η is indeed in an earlier batch under $\boldsymbol{p}$ than the batch agent ζ is in.

Claim 4. $t_\eta^d(\boldsymbol{p}) < t_\zeta^d(\boldsymbol{p})$.

Proof. By the global FIFO property of $\boldsymbol{p}$ stated in Theorem 3(i), the choice of $\zeta \in S_1$ enforces that ζ reaches the final destination d no earlier than any agent in S_1. Hence, agent ζ reaches d no earlier than any one of the $D+1$ agents in S_1' chosen from S_1.

Observe (by the definition of D) that the $D+1$ agents in S_1' cannot reach d at the same time. So the earliest one, agent η as per our selection, reaches d earlier than the latest one, and thus earlier than ζ. ☐

Next, we consider the scenario that when agent ζ arrives at o', he switches his subroute in G_1 to a best possible o'-d' path in G_2 to reach d', with all other agents still following their paths as in $\boldsymbol{p}$, for which we can upper bound ζ's arrival time at d' by $r_0 + L + |S_2|$. So we will focus on G_2. To facilitate analysis, we construct an auxiliary ADR game Γ' on G_2 as follows: The agent set Δ' consists of ζ and all the agents who pass through G_2 under $\boldsymbol{p}$. Each agent of Δ' enters G_2 from o' at the same time as he reaches o' under $\boldsymbol{p}$, and chooses an o'-d' path in G_2 as his route in Γ'. Game Γ' follows the same ADR model as game Γ_1, Γ_2, or Γ_3 in which $\boldsymbol{p}$ is considered. We define agent priorities in Γ' as follows.

- For Γ_1 and Γ_2, the global agent-priority ordering on Δ' is the restriction of the global agent-priority list of Δ to Δ'.
- For Γ_3, the original agent-priority ordering among agents of Δ' who reach o' simultaneously is the restrictions of that in Γ_3 when $o' = o$; it is determined by the edge priority on $E_{o'}^-$ (i.e., an agent who comes from an edge in $E_{o'}^-$ with a higher priority has a higher original priority) when $o' \neq o$.

Recalling Section 5.1, in the cases of Γ_1 and Γ_2 the ADR model requires that agents entering the network earlier have higher priorities than later ones. Since $\boldsymbol{p}$ is a UFR equilibrium in these cases, Lemma 2 guarantees that the global agent-priority ordering defined above for Γ' satisfies the modelling requirement. It is clear that the restriction of $\boldsymbol{p}$ to G_2, denoted as $\boldsymbol{p}|_{G_2}$, is a partial path profile of agents in $\Delta' \backslash \{\zeta\}$. In any case, we show in Appendix D that ζ has a best response to $\boldsymbol{p}|_{G_2}$, an o'-d' path denoted as P_ζ^*, such that under the routing $\boldsymbol{q} := (P_\zeta^*, \boldsymbol{p}|_{G_2})$ of game Γ', agent ζ can reach d' no later than any later-coming agents.

Claim 5. *Let $A_{\zeta,\boldsymbol{q}}$ denote the set of agents who enter G_2 later than ζ (i.e., after time r) under $\boldsymbol{q}$. Then no agent in $A_{\zeta,\boldsymbol{q}}$ can reach d' (i.e., exit G_2) earlier than agent ζ under $\boldsymbol{q}$.*

To bound ζ's arrival time at d' under $\boldsymbol{q}$ by $r_0+L+|S_2|$, we only need to consider the case where agent ζ is still inside G_2 at time r_0. We will apply Theorem 1, whose condition has been satisfied as Claim 5 states, to bound the time when agent ζ exits G_2 under $\boldsymbol{q}$. To this end, in addition to $A_{\zeta,\boldsymbol{q}}$ defined in Claim 5, we define $B_{\zeta,\boldsymbol{q}}(r_0)$ as the set of agents other than ζ who enter G_2 no later than agent ζ and are still inside G_2 at time r_0 under $\boldsymbol{q}$. The following claim relates the set $B_{\zeta,\boldsymbol{q}}(r_0)$ of agents in Γ' to the set S_2 of agents in the original game Γ_1, Γ_2, or Γ_3.

Claim 6. $B_{\zeta,\boldsymbol{q}}(r_0)\subseteq S_2$.

Proof. It suffices to show that: all agents exiting G_2 by time r_0 under $\boldsymbol{p}$ (if any) also exit G_2 by time r_0 under $\boldsymbol{q}$, namely, $t_h^{d'}(\boldsymbol{q})\le r_0$ holds for all $h\in\Delta'\backslash\{\zeta\}$ with $t_h^{d'}(\boldsymbol{p})\le r_0$. Recall from the no-overtaking property (Theorem 3(ii)) that ζ cannot affect the arrival times of any agent in an earlier batch by switching his own path. Suppose on the contrary that there is an agent h such that $t_h^{d'}(\boldsymbol{p})\le r_0$ and $t_h^{d'}(\boldsymbol{q})>r_0$. Since $\boldsymbol{q}$, when restricted to $\Delta'\backslash\{\zeta\}$, is a restriction of $\boldsymbol{p}$ to G_2, the different arrival times of h at d' under $\boldsymbol{p}$ and $\boldsymbol{q}$ imply that agent h's arrival time at d' is affected by ζ's unilateral change. So it must be the case that under $\boldsymbol{p}$, agent h reaches the final destination d no earlier than ζ, and hence later than agent η (by Claim 4). However, recalling Claim 3, under $\boldsymbol{p}$ agent η reaches d' later than $r_0\ge t_h^{d'}(\boldsymbol{p})$, i.e., later than h, showing a contradiction to the global FIFO property of $\boldsymbol{p}$ (Theorem 3 (i)). □

We are now able to bound agent ζ's arrival time at d' under routing $\boldsymbol{q}$ of Γ'.

Claim 7. $t_\zeta^{d'}(\boldsymbol{q})\le r_0+L+|S_2|$.

Proof. Combining Claims 5 and 6, we apply Theorem 1 directly to deduce that under $\boldsymbol{q}$, agent ζ exits G_2 no later than $r_0+L_{o'd'}+|B_{\zeta,\boldsymbol{q}}(r_0)|\le r_0+L+|S_2|$. □

Now turning back to the game in G, let us think of the routing that results from $\boldsymbol{p}=(P_i)_{i\in\Delta}$ by an unilateral deviation of agent ζ to o-d path $\hat{P}_\zeta=P_\zeta[o,o']\cup P_\zeta^*\cup P_\zeta[d',d]$, under which he first follows $P_\zeta[o,o']$ to o', then switches to G_2 at o' and follows P_ζ^* to d', and finally follows $P_\zeta[d',d]$ to d. The resulting path profile $(\hat{P}_\zeta,\boldsymbol{p}_{-\zeta})$ is denoted as $\hat{\boldsymbol{p}}$. Now we are ready to wrap up the proof of Lemma 3.

Proof of Lemma 3. Recall from Claim 4 that under $\boldsymbol{p}$, agent η is in an earlier batch and agent ζ is in a later batch. Thus by the no-overtaking property stated in Theorem 3(ii), we have $t_\eta^{d'}(\hat{\boldsymbol{p}})=t_\eta^{d'}(\boldsymbol{p})$, as ζ cannot affect η's arrival times, and $t_\eta^{d'}(\hat{\boldsymbol{p}})\le t_\zeta^{d'}(\hat{\boldsymbol{p}})$, as η preempts ζ no matter how the latter chooses his route through d'.

On the other hand, under the contradiction assumption $n_1>D(L+n_2+1)$, Claims 3 and 7 combine into $t_\eta^{d'}(\boldsymbol{p})\ge r_0+\frac{|S_1|-(D+1)}{D-1}>r_0+L+|S_2|\ge t_\zeta^{d'}(\boldsymbol{q})$. Note that $\boldsymbol{q}$ is the restriction of $\hat{\boldsymbol{p}}$ on G_2, giving $t_\zeta^{d'}(\hat{\boldsymbol{p}})=t_\zeta^{d'}(\boldsymbol{q})$. In turn $t_\eta^{d'}(\hat{\boldsymbol{p}})=t_\eta^{d'}(\boldsymbol{p})>t_\zeta^{d'}(\boldsymbol{q})=t_\zeta^{d'}(\hat{\boldsymbol{p}})$ shows a contradiction to the no-overtaking property that $t_\eta^{d'}(\hat{\boldsymbol{p}})\le t_\zeta^{d'}(\hat{\boldsymbol{p}})$. □

Remark 3. Consider extending the game to uniform-capacity networks in which every edge has an integer capacity c. The buffer selection rules in Γ_1 and Γ_2 work directly for the capacitated case, while the tie-breaking rule in Γ_3 needs to be refined by splitting edges and specifying priorities among the splitted edges (see [4]). For the capacitated setting, the inequality in Lemma 3 is generalized to $n_1\le cD(L+n_2+1)$.

A quick explanation of the generalization is obtained by reducing the problem to a unit-capacity network G' via splitting edges of the capacitated network G and mapping equilibrium $\boldsymbol{p}$ to a routing $\boldsymbol{p}'$ for the game on G' in the straightforward manner, which does not affect the agents' arrival times at any vertex. It is not hard to see that $\boldsymbol{p}'$ is an equilibrium of the game in G' (whose maximum in-degree is cD). The above remark becomes a corollary of Lemma 3. Alternatively, without resorting to edge splitting, one can derive the same result by slightly generalizing the definition of S_1' in the proof of Lemma 3 (see Appendix E).

6.2. Queue lengths. In this subsection, we employ a novel inductive argument to resolve the EQLP of ADR games Γ_1, Γ_2 and Γ_3 on series-parallel network G. We say that a cut of G is *full* at time r if the buffer of each edge in the cut is nonempty at time r. Let C and C' be two minimum (in terms of number of edges) *o-d* cuts in G. We say that C is *on the left* of C' if we can order the edges in the two cuts as $C = \{e_1, \ldots, e_k\}$ and $C' = \{e'_1, \ldots, e'_k\}$ such that for each $i \in \{1, \ldots, k\}$ with $e_i \neq e'_i$ there exists an *o-d* path in G which contains $\{e_i, e'_i\}$ and visits e_i before e'_i.

For any o'-d' series-parallel network H, let $\Xi(H)$ denote the leftmost minimum o'-d' cut of H. In particular, $|\Xi(H)|$ is the minimum cut size (i.e., capacity) of H. Removing cut $\Xi(H)$ from H leaves two connected components: the left one that contains o', and the right one that contains d'. Let H^l denote the graph obtained from the left component by attaching all edges in $\Xi(H)$, and let H^r denote the right component. So H is the edge-disjoint union of H^l and H^r. For any subgraphs H_1 and H_2 of G, we use $H_1 \cup H_2$ to denote the subgraph of G whose vertex (resp. edge) set is the union of the vertex (resp. edge) sets of H_1 and H_2. Recall that $\boldsymbol{p}$ denotes an arbitrary equilibrium of Γ_i, $i \in \{1,2,3\}$, on G. In this subsection, we additionally assume that

$$\text{under } \boldsymbol{p} \text{ at most } |\Xi(G)| \text{ agents enter } G \text{ at each integer time point.} \tag{6.1}$$

Since $|\Xi(G)|$ is bounded above by the in-degree of destination d and thus by the maximum in-degree D of G, it is instant that, for any $G_i \in \mathfrak{S}$, under $\boldsymbol{p}$ at most D agents enter G_i at every time point. In the following, we use $m := |E|$ to denote the total number of edges in G.

Definition 8. Let $\mathfrak{F}$ denote the set of series-parallel subnetworks $G_i \in \mathfrak{S}$ such that under $\boldsymbol{p}$ there exist finite integers F_i^l and F_i^r satisfying the following two conditions:

- $\Xi(G_i)$ is full as long as G_i^l accommodates more than F_i^l agents.
- G_i^r can accommodate at most F_i^r agents at any time.

We call vector (F_i^l, F_i^r) a *certificate* for $G_i \in \mathfrak{F}$. In the following, we use (F_i^l, F_i^r) to denote the certificate for $G_i \in \mathfrak{F}$ such that F_i^l and F_i^r are as small as possible.

First, we can see that set $\mathfrak{F}$ is nonempty since every single edge of G apparently belongs to $\mathfrak{F}$, a certificate of which is $(0,0)$. Second, given the certificate (F_i^l, F_i^r) for $G_i \in \mathfrak{F}$, we note that every vector $(\tilde{F}_i^l, \tilde{F}_i^r)$ with $(\tilde{F}_i^l, \tilde{F}_i^r) \geq (F_i^l, F_i^r)$ is a certificate for $G_i \in \mathfrak{F}$. In the following two lemmas, we are given $G_1, G_2 \in \mathfrak{F}$ with certificates (F_i^l, F_i^r), $i = 1,2$, such that G_i is an o_i-d_i series-parallel network for $i = 1,2$.

Lemma 4. *If $G_s \in \mathfrak{S}$ is the combination of G_1 and G_2 in series, then $G_s \in \mathfrak{F}$ with certificate (F_s^l, F_s^r) satisfying* $\max\{F_s^l, F_s^r\} \leq m(F_1^l + F_1^r + F_2^l + F_2^r + 1)$.

Proof. The series combination $G_s \in \mathfrak{S}$ implies that $D \leq m-1$. By symmetry, suppose w.l.o.g. that G_s is obtained from G_1 and G_2 by merging d_1 with o_2. We distinguish between two cases.

In the case of $|\Xi(G_1)| \leq |\Xi(G_2)|$, clearly $\Xi(G_s)$ is just $\Xi(G_1)$, $G_s^l = G_1^l$ and $G_s^r = G_1^r \cup G_2^l \cup G_2^r$. To see $G_s \in \mathfrak{F}$, we prove it has $(F_1^l, F_1^r + F_2^l + |\Xi(G_1)| + F_2^r)$ as a certificate. Since $G_1, G_2 \in \mathfrak{F}$, it suffices to show that the number of agents $G_1^r \cup G_2^l$ can accommodate is upper bounded by $F_1^r + F_2^l + |\Xi(G_1)|$ at any time. Since G_1^r can accommodate at most F_1^r agents, and $\Xi(G_2)$ is full as long as G_2^l accommodates more than F_2^l agents, we deduce that $\Xi(G_2)$ is full as long as $G_1^r \cup G_2^l$ accommodates more than $F_1^r + F_2^l$ agents. Consider any time point t such that $G_1^r \cup G_2^l$ accommodates at most $F_1^r + F_2^l$ agents at time t, and more than $F_1^r + F_2^l$ agents at time $t+1$. Then at time $t+1$, $G_1^r \cup G_2^l$ accommodates at most $F_1^r + F_2^l + |\Xi(G_1)|$ agents and $\Xi(G_2)$ is full. So $|\Xi(G_2)|$ agents exit $G_1^r \cup G_2^l$ at time $t+2$, while at most $|\Xi(G_1)|$ agents enter $G_1^r \cup G_2^l$. It follows from $|\Xi(G_1)| \leq |\Xi(G_2)|$ that the number of agents inside $G_1^r \cup G_2^l$ is non-increasing unless the number decreases below $F_1^r + F_2^l + 1$. Therefore, at any time $G_1^r \cup G_2^l$ can accommodate at most $F_1^r + F_2^l + |\Xi(G_1)|$ agents.

In the case of $|\Xi(G_1)| > |\Xi(G_2)|$, we see that $\Xi(G_s)$ is just $\Xi(G_2)$. We only need to show that $\Xi(G_2)$ is full as long as G_s^l $(= G_1 \cup G_2^l)$ accommodates more than $F_1^l + m(F_1^r + F_2^l)$ agents, which gives a certificate $(F_1^l + m(F_1^r + F_2^l), F_2^r)$ for $G_s \in \mathfrak{F}$. Suppose that t is a time point when $\Xi(G_2)$

is not full and $G_1 \cup G_2^l$ accommodates F agents. Next, we prove that $F \le F_1^l + (1+D)(F_1^r + F_2^l)$ holds, which implies the desired conclusion because $D \le m-1$. We only need to consider the case $F > F_1^l + F_1^r + F_2^l$. Since $G_2 \in \mathfrak{F}$, that $\Xi(G_2)$ is not full implies that G_2^l accommodates at most F_2^l agents at time t. It follows from $G_1 \in \mathfrak{F}$ that at time t, $G_1^r \cup G_2^l$ accommodates at most $F_1^r + F_2^l$ agents, and thus G_1^l accommodates at least $F - (F_1^r + F_2^l) > F_1^l$ agents. Let t' be the latest time before t when G_1^l accommodates at most F_1^l agents. Recall that under $\boldsymbol{p}$ at most D agents enter G_s at every time point. So from $t'+1$ to t at most $D(t-t')$ agents can enter G_1^l, and we have

$$F - (F_1^r + F_2^l) - F_1^l \le (t-t')D.$$

Since, by the choice of t', at times $t'+1, t'+2, \ldots, t$, there are more than F_1^l agents in G_1^l, cut $\Xi(G_1)$ is full at all these consecutive $t-t'$ time points. Therefore, at time t, the number of agents inside $G_1^r \cup G_2^l$ is at least

$$(t-t')(|\Xi(G_1)| - |\Xi(G_2)|) \ge t - t' \ge (F - F_1^l - F_1^r - F_2^l)/D.$$

On the other hand, recall that there are at most $F_1^r + F_2^l$ agents inside $G_1^r \cup G_2^l$ at time t. Now $(F - F_1^l - F_1^r - F_2^l)/D \le F_1^r + F_2^l$ gives $F \le F_1^l + (1+D)(F_1^r + F_2^l)$. The lemma is proved. □

Lemma 5. *If $G_p \in \mathfrak{G}$ is the combination of G_1 and G_2 in parallel, then $G_p \in \mathfrak{F}$ with certificate (F_p^l, F_p^r) satisfying* $\max\{F_p^l, F_p^r\} \le (m+1)(m + F_1^l + F_1^r + F_2^l + F_2^r)$.

The parallel combination $G_p \in \mathfrak{G}$ implies that $L \le m-1$. It is clear that $\Xi(G_p) = \Xi(G_1) \cup \Xi(G_2)$, $G_p^l = G_1^l \cup G_2^l$, and $G_p^r = G_1^r \cup G_2^r$. Therefore, G_p^r accommodates at most $F_p^r \equiv F_1^r + F_2^r$ agents at any time. To prove $G_p \in \mathfrak{F}$, we only need to show that $\Xi(G_1) \cup \Xi(G_2)$ is full as long as $G_1^l \cup G_2^l$ accommodates more than a certain finite number of agents. Since $G_1, G_2 \in \mathfrak{F}$, it suffices to consider the time when one of G_1^l and G_2^l, say G_1^l, accommodates at most F_1^l agents. Suppose that G_2^l accommodates F agents at that time. It follows from Lemma 3 that G_1 accommodates at least $F/D - (L+1)$ agents. Since $G_1 \in \mathfrak{F}$, there are at least $F/D - L - 1 - F_1^r$ agents inside G_1^l. It follows from $F/D - L - 1 - F_1^r \le F_1^l$ that $F \le D(L + 1 + F_1^l + F_1^r)$, and $G_1^l \cup G_2^l$ accommodates at most $F_1^l + F \le F_1^l + m(m + F_1^l + F_1^r)$ agents. Thus, if $\Xi(G_1) \cup \Xi(G_2)$ is not full, it must be the case that G_p^l accommodates at most $F_p^r \equiv (m+1)(m + F_1^l + F_1^r + F_2^l + F_2^r)$ agents. □

Set $F \equiv \max\{F_1^l, F_1^r, F_2^l, F_2^r\}$. Then G_1 and G_2 have (F, F) as a common certificate for their membership in $\mathfrak{F}$. It follows from Lemmas 4 and 5 that the series or parallel combination of G_1 and G_2 whichever belongs to $\mathfrak{G}$ has $(8m(m+F), 8m(m+F))$ as a certificate for its membership in $\mathfrak{F}$. Recall that all m edges in G belong to $\mathfrak{F}$ with certificates $(0,0)$, and G is obtained from the edges by performing $m-1$ series or parallel combinations. We deduce that $((8m)^m, (8m)^m)$ is a certificate for $G \in \mathfrak{F}$ since $8m(m + 8m(m + 8m(\cdots + 8m(m + 8m \cdot m)\cdots) = m\sum_{h=1}^{m-1}(8m)^h \le (8m)^m$. Now we are ready to present and prove the main result of this section.

Theorem 4. *Let G be a series-parallel network with m edges. If $|\Delta_t| \le |\Xi(G)|$ for all $t \ge 1$, then, for any UFR equilibrium of Γ_1 or Γ_2, or any NE of Γ_3, the number of agents in G at any time is upper bounded by a constant at most $(12m)^m$.*

Proof. We may assume $m \ge 2$. Notice from $|\Delta_t| \le |\Xi(G)|$ that the equilibrium, which we write as $\boldsymbol{p}$, satisfies (6.1). Therefore, we may define $\mathfrak{F}$ w.r.t. this $\boldsymbol{p}$, and obtain $G \in \mathfrak{F}$ with certificate $((8m)^m, (8m)^m)$ as argued above. So, under the equilibrium $\Xi(G)$ is full as long as G^l accommodates more than $(8m)^m$ agents, and G^r can accommodate at most $(8m)^m$ agents at any time. Similar to the argument used in the proof of Lemma 4, we consider any time t such that G^l accommodates at most $(8m)^m$ agents at time t, and more than $(8m)^m$ agents at time $t+1$. Then G^l accommodates at most $(8m)^m + |\Delta_{t+1}| \le (8m)^m + |\Xi(G)|$ agents and $\Xi(G)$ is full at time $t+1$. Thus $|\Xi(G)|$ agents exit G^l at time $t+2$ while $|\Delta_{t+2}|$ $(\le |\Xi(G)|)$ agents enter G^l. It follows that the number of agents inside G^l is non-increasing unless the number decreases below $(8m)^m + 1$. Therefore, at any time G^l can accommodate at most $(8m)^m + |\Xi(G)|$ agents, and G can accommodate at most $2(8m)^m + m$ agents. Since $m \ge 2$, the result follows. □

If we take the average residence time of all agents as a measure of the social cost as in [28], then the boundedness of the queue lengths implies that the PoA of NEs in Γ_3 and the PoA for the UFR equilibria in Γ_1 and Γ_2 are also bounded.

Remark 4. Further to Remark 3, the boundedness result in Theorem 4 can be generalized to uniform-capacity series-parallel networks with population being bounded by $(12cm)^{cm}$, where c is the uniform edge capacity.

7. Concluding remarks.

In this paper, we have studied the problems of bounding agents' residence times for a broad class of atomic dynamic routings, where the buffer regulation rules are flexible and the network may contain multiple origins and multiple destinations. As an application of the obtained results, we have established that for three classes of atomic dynamic routing games that appear in the recent literature, the residence times of selfish agents (or equivalently queue lengths) at equilibrium are all upper bounded by a constant depending only on the network, provided the network is series-parallel and the number of incoming agents at each time point does not exceed the network capacity. Our results generalize those in [24] and [28] on bounding residence times and answering the EQLP. The token techniques explored in this paper have exhibited capabilities to circumvent direct analysis of complicated chain effects in atomic dynamic routings and may have the potential to serve as a universal tool for a large class of similar problems.

Two interesting directions deserve future research efforts. (i) Theorem 2 is more general than Theorem 1 in that it allows for uniform capacities instead of unit ones. However, while agents may enter the network over time in Theorem 1, they are inside the network at the very beginning in Theorem 2. Are we able to extend Theorem 2 further to the more general scenario in which agents enter the network over time, edges have uniform or nonuniform capacities, and there may be multiple origins and multiple destinations? To accomplish these tasks, we may need to design more advanced token techniques by combining the advantages of both the multi-token and single-token techniques. The advantage of the multi-token technique comes from the large number of tokens (which may do a better job in "simulating" simultaneous movements of multiple agents), while its disadvantage lies in the fact that the movements of the tokens are relatively rigid. In contrast, the advantage of the single-token technique comes from the fact that the token's movement is more flexible (which contributes to more concise analysis), and its weakness lies in the limited capability of a single token in keeping an eye on the whole picture. Designing a more systematic token technique may also help tackle other complicated problems. (ii) Concerning the EQLP, are we able to extend Theorem 4 from series-parallel networks to more general ones or from UFR equilibria to general NEs for games Γ_1 and Γ_2? In attempts to tackle more general network topologies, the following difficulties make our key ideas hard to generalize. First, severe population imbalance between two "parallel" subnetworks may happen in general topology. For example, consider two internally vertex-disjoint subpaths $P_1[o,v]$ and $P_2[o,v]$ from origin o to a non-destination vertex v. In a general network, it is possible that $P_1[o,w]$ is very congested but $P_2[o,w]$ is empty, since the agents using some edges in $P_1[o,w]$ do not necessarily pass through v. In contrast, such a situation is prohibited by the series-parallel structure. Second, in a general network, there may not be any well-defined "leftmost" minimum cut. It is unclear which minimum cut must be full of passing agents once there are too many agents in the network, and which part of the network will always accommodate a limited number of agents. Thus tackling the EQLP for more general networks asks for new game theoretical and graphic insights and techniques.

APPENDICES

Appendix A: The partial order on single-token dynamics. In this section, we provide more details on the partial order defined in Definition 3.

Proof of Claim 2. First it is evident that the binary relation is anti-symmetric. Suppose dynamic $(P;t(e_1),t(e_2),\ldots,t(e_k))$ is smaller than $(P';t'(e'_1),t'(e'_2),\ldots,t'(e'_{k'}))$, and the latter is smaller than $(P'';t''(e''_1),t''(e''_2),\ldots,t''(e''_{k''}))$. Then there exist i and i' such that the following two conditions are satisfied: (i) $t(e_i)<t'(e'_i)$, the first $i-1$ edges (if any) on P are identical to the corresponding ones on P', and the leaving times from the buffer on each of these edges are identical for both dynamics. (ii) $t'(e'_j)<t''(e''_j)$, the first $j-1$ edges (if any) on P' are identical to the corresponding ones on P'', and the leaving times from the buffer on each of these edges are identical for both dynamics. If $i\le j$, then $t(e_i)<t'(e'_i)\le t''(e''_i)$, and for all $k<i$ (if any), $e_k=e'_k=e''_k$ and $t(e_k)=t'(e'_k)=t''(e''_k)$. If $i>j$, then $t(e_j)=t'(e'_j)<t''(e''_j)$, and for all $k<j$ (if any), $t(e_k)=t'(e'_k)=t''(e''_k)$. This shows that the binary relation is transitive. □

This partial order is very similar to a *lexicographic* order, which is *linear*, i.e., a partial order such that all pairs of elements are comparable. In fact, we show in the following that this order is linear when $c=1$. It is not the case, however, when $c\ge 2$, because multiple agents may leave a buffer simultaneously, making it impossible to compare the corresponding legal dynamics.

Remark 5. The partial order defined in Definition 3 is linear if $c=1$.

Proof. We prove that any two different legal dynamics are comparable. Let $D=(P;t(e_1),\ldots,t(e_k))$ and $D'=(P';t'(e'_1),\ldots,t'(e'_{k'}))$ be two different legal dynamics. According to our definition of legal dynamic (Definition 2), we know $e_1=e'_1=f$ and $t(e_k)=t'(e'_{k'})=\tau_\zeta-1$ (note that k is not necessarily equal to k'). Suppose that $t(e_1)=t'(e'_1)$. It must be the case that the token is held by the same agent when passing the starting edge f under D and D'. It follows that $e_2=e'_2$. An iterative argument shows that there must exist an $i\ge 2$ such that $e_1=e'_1,\ldots,e_i=e'_i$, $t(e_1)=t'(e'_1),\ldots,t(e_{i-1})=t'(e'_{i-1})$, and $t(e_i)\ne t(e'_i)$, as otherwise $t(e_k)=\tau_\zeta-1=t'(e'_{k'})$ would imply $D=D'$, a contradiction. □

Appendix B: A simpler proof for short path routing.

Proposition 1 (**[24]**). *Given an ADR profile on G_c, suppose all the $|\Delta|$ agents enter G_c at the same time and each agent follows a shortest origin-destination path. Then the residence time of any agent ζ is at most $l_\zeta+\lfloor(|\Delta|-1)/c\rfloor$, where l_ζ is the length of a shortest o_ζ-d_ζ path in G.*

Proof. To prove the proposition, we still use the single-token technique and its analysis developed in Section 4 under the additional condition that the token *can only move along shortest o_ζ-d_ζ paths*. That is to say, in each legal dynamic $D=(P;t(e_1),t(e_2),\ldots,t(e_k))$, the path $P=e_1e_2\ldots e_k$ can only be a shortest o_ζ-d_ζ path (with length $k\equiv l_\zeta$). Similarly, there still exists a minimal legal dynamic $D^*=(P^*;t(e^*_1),t(e^*_2),\ldots,t(e^*_k))$. Since P^* is a shortest o_ζ-d_ζ path with length l_ζ, we are left to prove that under D^*, the token is never delayed by ζ and can be delayed by another agent for at most once. The proof is almost the same as that of Theorem 2, except for a small but critical part concerning shortest paths.

Suppose on the contrary that the token is delayed twice by an agent η on edges e^*_i and e^*_j, where $1\le i<j\le k$, and u_i,u_j are tail vertices of e^*_i,e^*_j, respectively. Let P_η be η's route in the given ADR profile. Since all agents follow their shortest origin-destination paths, the sub-path $P^*[u_i,u_j]$ of P^* and sub-path $P_\eta[u_i,u_j]$ of P_η are both shortest u_i-u_j paths in G. It follows that path $P':=P^*[o_\zeta,u_i]\cup P_\eta[u_i,u_j]\cup P^*[u_j,d_\zeta]$ is a shortest o_ζ-d_ζ path. This allows us to modify D^* to obtain a smaller legal dynamic by letting agent η hold the token when he moves along the sub-path $P_\eta[u_i,u_j]$, which contradicts the minimality of D^*. □

Note that the basic intuitions behind our proof and that of Mansour and Pattshamir [24] are similar: trying to upper bound the waiting time of the token (or their "train conductor") by $\lfloor(|\Delta|-1)/c\rfloor$. However, our proof relies on the concept of legal dynamics (see Definition 2) and the partial order among them (see Definition 3 and Claim 2), while their proof is mainly based on the concept of "time path" (different from our legal dynamic) on shortest paths and the process of iteratively "switching" the time path. In fact, Theorem 2 can be considered as a generalization of Proposition 1 from shortest path networks to more general ones. It seems unlikely to derive Theorem 2 by using their "time path" analysis. Particularly, Mansour and Pattshamir [24] listed generalizing their result as an open problem.

Remark 6. Due to the shortest-path restrictions, the underlying network of the ADR investigated in Proposition 1 needs not be acyclic for validating the bound of $l_\zeta+\lfloor(|\Delta|-1)/c\rfloor$ even in the case of $c=1$.

Appendix C: UFR equilibria. Given any path profile $\boldsymbol{q}=(Q_i)_{i\in\Delta}$ and any agent subset $S\subseteq\Delta$, we use $\boldsymbol{q}_S=(Q_i)_{i\in S}$ to denote the partial path profile of agents in S. In particular, $\boldsymbol{q}_\emptyset$ represents a null path profile.

It turns out that UFR equilibria in Γ_1 and Γ_2 are the outcomes of the following algorithmic construction. Given an instance of game Γ_1 or Γ_2, we assume without loss of generality that the agents $1,2,\ldots$ in Δ are ordered in their decreasing priorities. A path profile $\boldsymbol{p}=(P_i)_{i\in\Delta}$ can be constructed by the following iterative adapted-shortest-path (IASP) algorithm [28]. Initially, P_1 is a shortest o-d path in G. Iteratively, for each $i=2,3,\ldots$, considering routing of the first i agents, and assuming that agents $1,\ldots,i-1$ choose routes $P_1,\ldots,P_{i-1}$, respectively, the algorithm finds (e.g., using a slight modification of Dijkstra's algorithm) an o-d path P_i along which agent i is able to reach each vertex of the path as early as possible. For convenience, we call such a path as an *earliest-arrival o-d path based on* $(P_1,\ldots,P_{i-1})$.

For every nonnegative integer i, let $[i]$ denote the set of positive integers no larger than i, As has been argued in [28], for each $i\geq 1$, as long as all agents in $[i]$ follow their routes in $\boldsymbol{p}_{[i]}=(P_1,\ldots,P_i)$ computed by the IASP algorithm, no agent $j\in\Delta\setminus[i]$ can preempt any agent in $[i]$. (Their arguments work for both Γ_1 and Γ_2 because the FIFO principle has never played any role in proving the non-overtaking.) Formally, for any agent indices i,j and agent subset S with $j\in S\subseteq\Delta\setminus[i-1]$, any vertex $v\in P_i$, and any partial path profile $(Q_h)_{h\in S}$, it holds that

$$t_i^v(\boldsymbol{p}_{[i]},(Q_h)_{h\in S\setminus\{i\}})=t_i^v(\boldsymbol{p}_{[i]})\leq t_j^v(\boldsymbol{p}_{[i-1]},(Q_h)_{h\in S}),\tag{C.1}$$

where $\boldsymbol{p}_{[0]}$ denotes the null path profile. In particular, we have $t_i^v(\boldsymbol{p}_{[i]})=\min_{R_i\in\mathscr{P}}t_i^v(\boldsymbol{p}_{[i-1]},R_i)=\min_{(R_j)_{j\in\Delta\setminus[i-1]}}t_i^v(\boldsymbol{p}_{[i-1]},(R_j)_{j\in\Delta\setminus[i-1]})$.

Lemma 6 (**[28]**). *Let $\boldsymbol{p}=(P_h)_{h\in\Delta}$ be computed by IASP algorithm.*
(i) *For every agent i, if all agents in $[i-1]$ follow their routes in $\boldsymbol{p}_{[i-1]}$, and i follows P_i, then no matter how other agents choose their routes, none of them can preempt i.*
(ii) *Under $\boldsymbol{p}$, no agent with a lower priority can preempt any agent with a higher priority.*

Now for each $i\geq 1$, considering the situation where all agents in $[i-1]$ follow $\boldsymbol{p}_{[i-1]}$ and agent i takes route $Q_i\in\mathscr{P}$, for an arbitrary vertex $v\in Q_i$, we compare i's arrival time at v under partial routing $(\boldsymbol{p}_{[i-1]},Q_i)$ and that under a routing $(\boldsymbol{p}_{[i-1]},Q_i,(R_j)_{j\in S})$ when more agents in any set $S\subseteq\Delta\setminus[i]$ with any routes R_j, $j\in S$, are added. Since no agent in S can preempt any agent in $[i-1]$, the arrival time of every agent in $[i-1]$ at every vertex is the same under the two routings. It follows that the addition of routes R_j, $j\in S$, cannot speed up i's travel, i.e.,

$$t_i^v(\boldsymbol{p}_{[i-1]},Q_i)\leq t_i^v(\boldsymbol{p}_{[i-1]},Q_i,(R_j)_{j\in S}).\tag{C.2}$$

Lemma 7. *For Γ_1 and Γ_2, the set $\mathscr{E}$ of UFR equilibria is identical to the set $\mathscr{A}$ of path profiles computed by all possible implementations of the IASP algorithm.*

Proof. The inclusion of $\mathscr{A} \subseteq \mathscr{E}$ has been argued in [28] (see the proof of Lemma 1 over there), where the FIFO principle does not play any role. To see the reverse inclusion, suppose for a contradiction that there is a UFR equilibrium $\boldsymbol{q} = (Q_h)_{h\in\Delta} \notin \mathscr{A}$. Let i be the smallest agent index such that $\boldsymbol{q}_{[i-1]}$ is a partial path profile found by some implementation of the IASP algorithm and Q_i is not an earliest-arrival o-d path based on $\boldsymbol{q}_{[i-1]}$. Let $v \in Q_i$ be the closest vertex to d such that $t_i^v(\boldsymbol{q}_{[i]}) > \min_{R_i\in\mathscr{P}} t_i^v[\boldsymbol{q}_{[i-1]}, R_i]$. Using a modification of Dijkstra's algorithm, one can compute an o-v path R such that if agents in $[i-1]$ follow $\boldsymbol{q}_{[i-1]}$, the earliest arrival time of i at every vertex of R, among all partial routings $(\boldsymbol{q}_{[i-1]}, R_i)$, is realized by i following R. The choice of v implies that $Q_i' := R \cup Q_i[v,d]$ is an earliest-arrival o-d path based on $\boldsymbol{q}_{[i-1]}$, and partial routing $(\boldsymbol{q}_{[i-1]}, Q_i')$ could be produced by some implementation of the IASP algorithm. It is evident from (C.1) that $t_i^v(Q_i', \boldsymbol{q}_{-i}) = t_i^v(Q_i', \boldsymbol{q}_{[i-1]}) = \min_{R_i\in\mathscr{P}} t_i^v[\boldsymbol{q}_{[i-1]}, R_i] < t_i^v(\boldsymbol{q}_{[i]})$. On the other hand, by the choice of i, inequality (C.2) implies $t_i^v(\boldsymbol{q}_{[i]}) \le t_i^v(\boldsymbol{q})$. The contradiction $t_i^v(Q_i', \boldsymbol{q}_{-i}) < t_i^v(\boldsymbol{q})$ to the UFR property (see Definition 5) proves the lemma. □

The following no-overtaking property is an immediate corollary of Lemmas 6 and 7.

Lemma 8. *Let $\boldsymbol{p}$ be a UFR equilibrium of Γ_1 or Γ_2, and $\lambda \in \Delta$ be an arbitrary agent.*

(i) *Under $\boldsymbol{p}$, no agent with a lower priority can preempt an agent with a higher priority.*

(ii) *There is an o-d path $P_\lambda^* \in \mathscr{P}$ such that, if all agents with higher priorities than λ follow their routes in $\boldsymbol{p}$, and λ follow P_λ^*, then no matter how other agents choose their routes, none of them can preempt λ (at any vertex of P_λ^*). Q.E.D.*

Appendix D: The earliest-arrival best response. First, we study agents' special best responses in game Γ_3, in which we are given a fixed agent ζ and a fixed partial path profile $\boldsymbol{p}_{-\zeta} = (P_j)_{j\in\Delta\setminus\{\zeta\}}$ of all agents other than ζ. For each vertex $v \in V$ and agent $i \in \Delta$, define

$$\tau_i^v := \min_{P_\zeta \in \mathscr{P}} \{t_i^v(P_\zeta, \boldsymbol{p}_{-\zeta})\}$$

as the earliest time at which agent i can reach v when only agent ζ can change his path (noting that $\boldsymbol{p}_{-\zeta}$ has been fixed). It has been shown in Section 4.3 of [3] that the best response of agent ζ defined below exists, and can be computed efficiently.

Definition 9 (Best response). An o-d path $P_\zeta^* \in \mathscr{P}$ is called agent ζ's *earliest-arrival best response* (w.r.t. $\boldsymbol{p}_{-\zeta}$) if under routing $(P_\zeta^*, \boldsymbol{p}_{-\zeta})$, agent ζ reaches each vertex v of P_ζ^* the earliest (i.e., at time τ_ζ^v) among all o-d paths, and when there are more than one way for ζ to reach a vertex v of P_ζ^* at time τ_ζ^v, path P_ζ^* always uses an entering edge with the highest edge priority.

The following definition and lemma concerning "domination" can be found in Section 4.2 of [3].

Definition 10 (Domination). For every agent $j \in \Delta\setminus\{\zeta\}$ and vertex $v \in P_j$, we say agent ζ *dominates* agent j at vertex v if either (a) $\tau_\zeta^v < \tau_j^v$, or (b) $\tau_\zeta^v = \tau_j^v$ and there exists $P_\zeta \in \mathscr{P}$ such that $v \in P_\zeta$, $t_\zeta^v(P_\zeta, \boldsymbol{p}_{-\zeta}) = \tau_\zeta^v$ and, either $\epsilon_v^-(P_\zeta)$ has a priority higher than $\epsilon_v^-(P_j)$ (if $v \ne o$) or ζ has a higher original priority than j (if $v = o$).

Lemma 9 (**[3]**). *If ζ dominates agent $j \in \Delta\setminus\{\zeta\}$ at vertex $v \in P_j$, then ζ dominates j at all the vertices on the path $P_j[v,d]$.*

Lemma 10. *In game Γ_3, suppose that $\zeta \in \Delta$ is an arbitrary agent, and $\boldsymbol{p}_{-\zeta}$ is a partial path profile of all agents other than ζ. Then there exists $P_\zeta^* \in \mathscr{P}$ such that, under $(P_\zeta^*, \boldsymbol{p}_{-\zeta})$, agent ζ reaches d no later than any agent who enter G later than ζ.*

Proof. Let $\boldsymbol{p}^* := (P^*_\zeta, \boldsymbol{p}_{-\zeta})$ denote the routing in which agent ζ follows his earliest-arrival best response w.r.t. $\boldsymbol{p}_{-\zeta}$ as defined in Definition 9. Let j be an arbitrary agent who enters the network G later than ζ. It suffices to show that under $\boldsymbol{p}^*$, agent ζ reaches d no later than agent j. By Definition 10, it is apparent that agent ζ dominates agent j at the origin o. By Lemma 9, agent ζ dominates agent j at destination d. Since under $\boldsymbol{p}^*$, agent ζ makes his earliest-arrival best response w.r.t. $\boldsymbol{p}_{-\zeta} = \boldsymbol{p}^*_{-\zeta}$, he reaches d at time τ^d_ζ, which is no later than $\tau^d_j \le t^d_j(\boldsymbol{p}^*)$. □

We now prove Claim 5 concerning earliest-arrival best responses in games Γ_1, Γ_2 and Γ_3.

Proof of Claim 5. In the case of Γ_3, the claim is straightforward from Lemma 10. It remains to consider Γ_1 and Γ_2. Since $\boldsymbol{p}$ is a UFR equilibrium of the game on G, we note again from Lemma 8(i) that all agents in $A_{\zeta,\boldsymbol{q}}$, who are preempted by ζ at vertex o', have priorities lower than ζ in both games Γ_i ($i \in \{1,2\}$) and Γ'. Let Ω (resp. Ω') consist of agents in Δ (resp. Δ') whose priorities are higher than ζ. Then $A_{\zeta,\boldsymbol{q}} \subseteq \Delta' \setminus (\Omega' \cup \{\zeta\})$. Notice from Lemma 7 that $\boldsymbol{p}_\Omega$ is a partial path profile of the agents with the first $|\Omega|$ highest priorities computed by an implementation of the IASP algorithm for game Γ_i. It is easy to see that the restriction of $\boldsymbol{p}_\Omega$ to G_2, i.e., $(\boldsymbol{p}|_{G_2})_{\Omega'}$, is a partial path profile of the agents with the first $|\Omega'|$ highest priorities computed by an implementation of the IASP algorithm for game Γ'. Therefore, the IASP algorithm can compute a path profile $\boldsymbol{q}$ with $\boldsymbol{q}_{\Omega'} = (\boldsymbol{p}|_{G_2})_{\Omega'}$ for the game Γ'. By Lemma 7, we see that $\boldsymbol{q}$ is a UFR equilibrium of Γ'. In turn, Lemma 8(ii) provides ζ with an o'-d' path P^*_ζ in G_2 such that under routing $(\boldsymbol{q}_{\Omega'}, P^*_\zeta, (\boldsymbol{p}|_{G_2})_{\Delta' \setminus (\Omega' \cup \{\zeta\})}) = (P^*_\zeta, \boldsymbol{p}|_{G_2})$, no agent in $\Delta' \setminus (\Omega' \cup \{\zeta\}) \supseteq A_{\zeta,\boldsymbol{q}}$ can reach d' earlier than ζ. □

Appendix E: An alternative proof of Remark 3. We first note that all results in Section 5.2 and Appendices C and D remain valid for the game on the capacitated network. To prove Remark 3, similar to the proof of Lemma 3, we assume by contradiction that $n_1 > cD(L + n_2 + 1)$. For the generalized setting, we change the definition of S'_1 to be a set of $cD + 1$ agents in S_1 who reach d' as late as possible. The inequality in Claim 3 changes to $t^{d'}_\eta(\boldsymbol{p}) \ge r_0 + \frac{|S_1| - (cD+1)}{c(D-1)}$ since at most $c(D-1)$ agents in S_1 can reach d' at the same time. Claim 4 remains to hold becuase no $cD + 1$ agents can reach d at the same time. By the validity of statements in Appendices C and D, we still have Claim 5. As remarked at the beginning of Section 4, the unit-capacity assumption in Theorem 1 could be dropped without any effect on the claimed bound $r_0 + L_{o'd'} + |B_{\zeta,\boldsymbol{q}}(r_0)|$ of ζ's exiting time from G_2. Again, the no-overtaking property implies Claim 6 and therefore the same inequality $t^{d'}_\zeta(\boldsymbol{q}) \le r_0 + L + |S_2|$ as in Claim 7. Now combining $t^{d'}_\eta(\boldsymbol{p}) \ge r_0 + \frac{|S_1| - (cD+1)}{c(D-1)}$, $t^{d'}_\zeta(\boldsymbol{q}) \le r_0 + L + |S_2|$ and $n_1 > cD(L + n_2 + 1)$, we reach the same contradiction to the no-overtaking property as in the proof of Lemma 3.

Acknowledgments.

The first author was supported in part by the National Natural Science Foundation of China (NSFC) under grant numbers 71922003 and 71871009; the third author was supported in part by Ministry of Science and Technology of China under Grant number 2018AAA0101002, and by CAS under grant numbers ZDBS-LY-7008 and XDA27000000; the fourth author was supported in part by NSFC under grant number 11971046, and by YESS under grant number 2018QNRC001. The authors thank the editors and the review team for their helpful comments and suggestions.

References

[1] Busch C, Magdon-Ismail M, Mavronicolas M (2007) Universal bufferless packet switching. *SIAM Journal on Computing* 37(4):1139–1162.

[2] Busch C, Magdon-Ismail M, Mavronicolas M, Spirakis P (2006) Direct routing: Algorithms and complexity. *Algorithmica* 45(1):45–68.

[3] Cao Z, Chen B, Chen X, Wang C (2017) A network game of dynamic traffic. *Proceedings of the 2017 ACM Conference on Economics and Computation*, 695–696, EC '17 (ACM).

[4] Cao Z, Chen B, Chen X, Wang C (2021) Atomic dynamic flow games: Adaptive vs. nonadaptive agents. *Operations Research* URL http://dx.doi.org/10.1287/opre.2021.2105.

[5] Cidon I, Kutten S, Mansour Y, Peleg D (1995) Greedy packet scheduling. *SIAM Journal on Computing* 24(1):148–157.

[6] Cominetti R, Correa J, Larré O (2015) Dynamic equilibria in fluid queueing networks. *Operations Research* 63(1):21–34.

[7] Cominetti R, Correa J, Olver N (2017) Long term behavior of dynamic equilibria in fluid queuing networks. Eisenbrand F, Koenemann J, eds., *Integer Programming and Combinatorial Optimization*, 161–172 (Springer).

[8] Correa J, Cristi A, Oosterwijk T (2019) On the price of anarchy for flows over time. *Proceedings of the 2019 ACM Conference on Economics and Computation*, 559—-577, EC '19 (ACM).

[9] Fleischer L, Skutella M (2007) Quickest flows over time. *SIAM Journal on Computing* 36(6):1600–1630.

[10] Graf L, Harks T (2020) The price of anarchy for instantaneous dynamic equilibria. Chen X, Gravin N, Hoefer M, Mehta R, eds., *Web and Internet Economics*, 237–251 (Cham: Springer).

[11] Hall A, Hippler S, Skutella M (2007) Multicommodity flows over time: Efficient algorithms and complexity. *Theoretical Computer Science* 379(3):387–404.

[12] Harks T, Peis B, Schmand D, Tauer B, Koch LV (2018) Competitive packet routing with priority lists. *ACM Transactions on Economics and Computation* 6(1):4:1–4:26.

[13] Hendrickson C, Kocur G (1981) Schedule delay and departure time decisions in a deterministic model. *Transportation Science* 15(1):62–77.

[14] Hoefer M, Mirrokni VS, Röglin H, Teng SH (2011) Competitive routing over time. *Theoretical Computer Science* 412(39):5420–5432.

[15] Hoppe B, Tardos E (2000) The quickest transshipment problem. *Mathematics of Operations Research* 25(1):36–62.

[16] Horni A, Nagel K, Axhausen KW (2016) *The Multi-Agent Transport Simulation MATSim* (Ubiquity Press).

[17] Koch R, Peis B, Skutella M, Wiese A (2009) Real-time message routing and scheduling. Dinur I, Jansen K, Naor J, Rolim J, eds., *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, 217–230 (Springer).

[18] Koch R, Skutella M (2009) Nash equilibria and the price of anarchy for flows over time. Mavronicolas M, Papadopoulou VG, eds., *Algorithmic Game Theory*, 323–334 (Berlin, Heidelberg: Springer).

[19] Koch R, Skutella M (2011) Nash equilibria and the price of anarchy for flows over time. *Theory of Computing Systems* 49(1):71–97.

[20] Leighton F, Maggs B, Richa A (1999) Fast algorithms for finding o(congestion + dilation) packet routing schedules. *Combinatorica* 19:375–401.

[21] Leighton FT, Maggs BM, Rao SB (1994) Packet routing and job-shop scheduling ino (congestion+ dilation) steps. *Combinatorica* 14(2):167–186.

[22] Leighton T (1994) Methods for message routing in parallel machines. *Theoretical Computer Science* 128(1):31–62.

[23] Macko M, Larson K, Steskal Ľ (2013) Braess's paradox for flows over time. *Theory of Computing Systems* 53(1):86–106.

[24] Mansour Y, Patt-Shamir B (1993) Greedy packet scheduling on shortest paths. *Journal of Algorithms* 14(3):449–465.

[25] Ostrovsky R, Rabani Y (1997) Universal $O(\text{congestion} + \text{dilation} + \log^{1+\epsilon} N)$ local control packet switching algorithms. *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, 644–653, STOC '97 (ACM).

[26] Peis B, Skutella M, Wiese A (2010) Packet routing: Complexity and algorithms. Bampis E, Jansen K, eds., *Approximation and Online Algorithms*, 217–228 (Springer).

[27] Rabani Y, Tardos E (1996) Distributed packet switching in arbitrary networks. *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, 366–375, STOC '96 (ACM).

[28] Scarsini M, Schröder M, Tomala T (2018) Dynamic atomic congestion games with seasonal flows. *Operations Research* 66(2):327–339.

[29] Sering L, Koch LV (2019) Nash flows over time with spillback. *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, 935–945, SODA '19 (USA: SIAM).

[30] Sering L, Skutella M (2018) Multi-source multi-sink nash flows over time. Borndörfer R, Storandt S, eds., *Proceedings of the 18th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2018)*, 12:1–12:20 (Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik).

[31] Srinivasan A, Teo CP (2001) A constant-factor approximation algorithm for packet routing and balancing local vs. global criteria. *SIAM Journal on Computing* 30(6):2051–2068.

[32] Vickrey WS (1969) Congestion theory and transport investment. *The American Economic Review* 59(2):251–260.

[33] Werth T, Holzhauser M, Krumke S (2014) Atomic routing in a deterministic queuing model. *Operations Research Perspectives* 1(1):18–41.