A mathematical framework for delay analysis in single source networks

Samitha Samaranayake, Axel Parmentier, Yiguang (Ethan) Xuan, Alexandre Bayen

***Abstract*—This article presents a mathematical framework for modeling heterogeneous flow networks with a single source and multiple sinks. The traffic is differentiated by the destination (i.e. Lagrangian flow) and different flow groups are assumed to satisfy the first-in-first-out (FIFO) condition at each junction. The queuing in the network is assumed to be contained at each junction node and spill-back to the previous junction is ignored. We show that our model leads to a well-posed problem for computing the dynamics of the system and prove that the solution is unique through a mathematical derivation of the model properties. The framework is then used to analytically prescribe the delays at each junction of the network and across any sub-path, which is one of the contributions of the article. This is a critical requirement when solving control and optimization problems over the network, such as system optimal network routing and solving for equilibrium behavior. In fact, the framework provides analytical expressions for the delay at any node or sub-path as a function of the inflow at any upstream node. Furthermore, the model can be solved numerically using a very simple and efficient feed forward algorithm. We demonstrate the versatility of the framework by applying it to two example networks, a single path of multiple bottlenecks and a diverge junction with complex junction dynamics.**

I. Introduction

Modeling and analysing the dynamics of network flows is an important problem that has applications in many different areas such as transportation planning [3, 6, 12], air traffic control [9, 16], communication networks [1, 2, 4, 5], processor scheduling [17] and supply chain optimization [10]. Flow models are crucial for understanding the response of networked systems under different boundary conditions, estimating the state of the system, measuring system performance under different tunable parameters and devising the appropriate control strategies for efficient operation of the system. For example, in transportation networks, flow models are used for traffic estimation [19], dynamic traffic assignment or demand response assessment [8], traffic signal control [7], ramp-metering control [13] and incident rerouting [15]. This article focuses on modeling heterogeneous (multi-path) physical flows through a network with a single source and multiple sinks with the specific objective of expressing the delays at each node of the network as a function of the boundary flows at the source. This can be a critical requirement when solving control and optimization problems over a network in cases where the flow entering the network is one of the direct or indirect control parameters of the system. For example, when trying to eliminate congestion at a critical node of the network by manipulating the boundary flows [14]. We present our model in the context of physical flow networks and particularly freeway transportation networks, which have the following physical requirements, but our results can be applied to any network that satisfies these properties: 1) link flows are capacity restricted, 2) the flow through each junction satisfies the first-in-first-out (FIFO) condition, and 3) there is no holding of flow, i.e. the flow through a junction is maximized subject to the FIFO condition.

While there is a vast literature on network flow propagation, particularly for various packet networks, a large majority of these dynamics models violate the FIFO and no holding requirements listed above, which are essential requirements in phyisical flow networks. Many models proposed for transportation network flows do infact satisfy these physical requirements [3, 6], but none of these models analytically prescribe the internal delays of the network as a function of the boundary flows. Therefore, a new framework is required for the problem that we consider.

Our approach can be summarized as follows. We assume that the traffic flow is differentiated by the destination of the flow (i.e. Lagrangian flow) and that the different flow groups satisfy the FIFO condition at each junction. The queuing in the network is assumed to be contained at each junction node and spill-back to the previous junction if occurs is ignored[1]. We show that our model leads to a well-posed ordinary differential equation for computing the dynamics of the network as a function of the boundary flows and prove that the solution is unique through a mathematical derivation of the model properties. The main benefit of this framework is the ability to analytically prescribe the delays at any junction in the network and across any sub-path as a function of the the boundary flows, which can be a important requirement when solving certain control and optimization problems, such as demand allocation problems, where the flow entering the network is one of the direct or indirect control parameters. This is

S. Samaranayake is with the Department of Civil and Environmental Engineering at the University of California, Berkeley CA 94720 e-mail: samitha@berkeley.edu

A. Parmentier is with the Centre d'Enseignement et de Recherche en Mathmatiques et Calcul Scientifique at cole des Ponts ParisTech, Cit Descartes - Champs sur Marne, 77455 Marne la Valle Cedex 2 e-mail : axel.parmentier@enpc.fr

Y. Xuan is with California Partners for Advanced Transportation Technology at the University of California, Berkeley CA 94720 e-mail: xuan.yiguang@path.berkeley.edu

A. Bayen is with the Department of Electrical Engineering and Computer Science, and Civil and Environmental Engineering at the University of California, Berkeley CA 94720 e-mail: bayen@berkeley.edu

[1]Spill back to the previous junction can be observed and flagged when it occurs. The primary goal of this model is for being used in optimization problems where (in most cases) a good solution will eliminate long spill backs.

achieved via the creation of a time mapping operator that maps the traffic flow at a given node at a given time to the corresponding flow at the origin of the network when that flow entered the network. We also show that this model can be solved numerically using a simple and efficient forward simulation approach. Finally, we demonstrate the application of the model by applying it to two example networks, a single path of multiple bottlenecks and a diverge junction with complex junction dynamics.

The article is organized as follows. Section 2 introduces the network properties and junction dynamics. Then section 3 formalizes the time mapping operator, shows the well-posedness of the problem and proves the uniqueness of the solution to this model. Section 4 demonstrates the practical application of the mathematical framework by showing that the off-ramp model posed by Newell [11] can be modeled using this framework. Section 5 concludes the article.

II. A POINT QUEUE MODEL FOR NETWORK FLOW

The traffic network with a single source is modeled as an arborescence[2]. The congestion at each bottleneck is modeled as a vertical queue that is located at the start of the bottleneck. Thus, the physical propagation of the queue forming at the bottleneck is not modeled. This modeling choice is only restrictive when the queue propagates upstream to the preceding junction, as the change in dynamics at the junction due to the queue is not taken into account, but the model is equivalent to a horizontal queuing model otherwise.

A. Network definitions

A node v denotes a junction in the network and V is the set of all nodes. A link $l = (v_l^{\text{in}}, v_l^{\text{out}})$ is a couple consisting of an origin node v_l^{in} and a destination node v_l^{out}, and L is the set of all links.

The congestion-free travel time on link l is denoted by T_l, an agent that enters link l at time t will exit link l at time $t + T_l$. The congestion-free travel time between nodes v_1 and v_2 is denoted by $T_{(v_1,v_2)}$, an agent that enters node v_1 at time t will reach node v_2 at time $t + T_{(v_1,v_2)}$

The set of incoming links to node v is denoted by L_v^{in}, the set of outgoing links from node v is denoted by L_v^{out} and the set of all links l connected to node v is denoted by L_v.

$$L_v^{\text{in}} = \{l \in L | v_l^{\text{out}} = v\},\ L_v^{\text{out}} = \{l \in L | v_l^{\text{in}} = v\} \quad (1)$$

A node v is a source if it admits no incoming link ($L_v^{\text{in}} = \emptyset$). A node v is a sink if it admits no exiting link ($L_v^{\text{out}} = \emptyset$). The set of sinks is denoted by S.

The set of nodes V and the set of links L compose a network. Due to the network being an arborescence, it contains a unique source indexed by v_0. For all nodes $v \in V\backslash\{v_0\}$, L_v^{in} is a singleton. The element of this singleton is called the *parent* node and is denoted by π_v: $L_v^{\text{in}} = \{(\pi_v, v)\}$.

We define a path $p_{(v_{\text{orig}}, v_{\text{dest}})}$ as a finite sequence of distinct nodes from an origin node v_{orig} to a destination node v_{dest} such that there is a link connecting each pair of subsequent nodes.

$p_{(v_{\text{orig}}, v_{\text{dest}})} = (v_{\text{orig}}, \cdots, v_{\text{dest}})$ s.t. $(\pi_{v_i}, v_i) \in L \quad \forall i \in p\backslash v_{\text{orig}}$

[2] An arborescence is a directed rooted tree where all edges point away from the root

There is a unique path from any source to any destination since the network is tree structured. For each sink s, let p_s be the path starting at the origin v_{orig} and ending at node $v_s = s$, and V_{p_s} be the sequence of nodes on path p_s. The set of paths P_v is the set of all paths p for which $v \in p$. The set of paths P_l is the set of all paths p for which $l \in p$.

$$P_v = \{p | v \in V_p\}\ ;\ P_l = \{p | v_l^{\text{in}} \in V_p \text{ and } v_l^{\text{out}} \in V_p\} \quad (2)$$

Remark 1. *The path sets P_l where l is a link in L_v^{out} form a partition of P_v*

$$P_v = \cup_{l \in L_v^{out}} P_l \quad (3)$$

□

B. Modeling the flow of agents

The traffic flow at a node is measured by counting the number of agents that pass through the node between an arbitrary initial time t_{initial} and any given time t.

For a node $v \in V\backslash v_0$ (that is not the source) and path $p \in P_v$, the arrival curve $A_v^p(t)$ gives the total number of agents on path p that arrive at node v during the time interval $(t_{\text{initial}}, t]$. Similarly, for a node $v \in V\backslash S$ (that is not a sink) and $p \in P_v$, the departure curve $D_v^p(t)$ gives the total number of agents on path p that leave node v during the time interval $(t_{\text{initial}}, t]$.

Remark 2. *The arrival curve $A_v^p(t)$ (resp. departure curve $D_v^p(t)$) also gives the agent number of the last agent on path p to arrive at (resp. leave) node v by t. Arrival and departure curves are monotonically increasing: if $t_1 < t_2$, $A(t_2) - A(t_1)$ (resp. $D_p(t_2) - D_p(t_1)$) is the total number of agents who arrive at (resp. pass) node v in the interval $(t_1, t_2]$, and is therefore non-negative.*

Definition 1. *Acceptable cumulative arrival and departure curves $\mathbf{A}(t_{initial}, t_{final}]$, $\mathbf{D}(t_{initial}, t_{final}]$*
Given times $t_{initial}$ and t_{final}, a function on $(t_{initial}, t_{final}]$ is an acceptable cumulative curve on $(t_{initial}, t_{final}]$ if it is continuous, piecewise C^1, and strictly increasing functions on $(t_{initial}, t_{final}]$.

The assumption that the cumulative curves are strictly increasing is made for mathematical convenience, but can be relaxed[3]. Cumulative curves are required to be C^1 in order to be able to define flows.

The outgoing flow λ_p^v at a node v is the piecewise continuous derivative of the departure curve D_p^v

$$\lambda_p^v = \frac{dD_p^v}{dt} \quad (4)$$

Remark 3. *Zero congestion-free travel time*
Let π_v, v be two consecutive nodes on path p. agents on path

[3] We could relax the assumption that the cumulative curves are strictly increasing and allow for monotonically increasing curves. However, this results in the time mapping function $T^{('\pi_v)}v$ introduced in section III-B being a correspondence instead of a function and makes the analysis significantly more complicated. Therefore, for mathematical convenience, we make the assumption that the cumulative curves are strictly increasing.

p leaving node v at time t arrive at node v at $t+T_{(\pi_v,v)}$*. For all links* (π_v, v) *and paths* $p \in P_v$*, without loss of generality we set the congestion-free travel time* $T_{(\pi_v,v)}$ *to be zero:* $T_{(\pi_v,v)} = 0$*. This implies that:*

$$D_p^{\pi_v} = A_p^v \qquad \forall l = (\pi_v, v) \in L, p \in P_v \tag{5}$$

This modeling choice is made purely for mathematical convenience, since the goal of this framework is to analyze delays in the network. The total travel time for each agent can be easily reconstructed a posteriori by adding the actual congestion-free travel time for each link of the path traveled by the agent.

Thus, for all links $(\pi_v, v) \in L$ and paths $p \in P$ we have:

$$\frac{dA_p^v}{dt} = \frac{dD_p^{\pi_v}}{dt} = \lambda_p^{\pi_v} \tag{6}$$

$$\frac{dD_p^v}{dt} = \lambda_p^v. \tag{7}$$

C. Queuing and diverge model

This section defines the model dynamics for queuing and the flow propagation through a junction, which will then lead to a definition of the feasible departure curves that the model admits.

The capacity $\mu_l(t)$ of a link l is the maximum flow that can enter the link from its input node v_l^{in} at time t. Road capacity may vary with time due to weather conditions, accidents, or other factors. Thus, capacity is a time varying quantity.

Requirement 1. *Capacity constrained flows*
The inflow entering a link is always no greater than the links capacity.

$$\sum_{p \in P_l} \lambda_p^{v_l^{in}}(t) \le \mu_l(t) \qquad \forall t, l \in L \tag{8}$$

If the flows arriving at a node v are larger than available outflow capacity, a queue will form at node v.

Definition 2. *Queue length* $n_{v,p}(t)$
We define the path queue length $n_{v,p}(t)$ *at node* v *as the number of agents on path* p *that arrive at node* v *by time* t *and are yet to depart node* v

$$n_{v,p}(t) = D_p^v(t) - A_p^v(t) \tag{9}$$

The total queue length $n_v(t)$ *at node* v *is the sum of the path queue lengths.*

$$n_v(t) = \sum_{p \in P_v} n_{v,p}(t) \tag{10}$$

Remark 4. *Let* $[D^v]^{-1}$ *be the inverse of the departure curve* D^v*. Since* D^v *is strictly increasing,* $t_k = [D^v]^{-1}(k)$ *gives the time at which agent number* k *leaves node* v*.*

Definition 3. *Delay in queue* v
We define $\delta_{v,p}(t)$ *as the delay encountered in queue* v *by the agent which entered the queue at time* t*.*

$$\begin{aligned} \delta_{v,p}(t) \quad &= [D_p^v]^{-1}\left(A_p^v(t)\right) - t \\ &= [D_p^v]^{-1}\left(D_p^{\pi_v}(t)\right) - t \end{aligned} \tag{11}$$

As D_p^v is continuous, piecewise C^1, and strictly increasing, its inverse is continuous, piecewise C^1 and strictly increasing. Thus, as $D_p^{\pi_v}$ is also continuous, piecewise C^1 and strictly increasing, the function $[D_p^v]^{-1} \circ D_p^{\pi_v}$ is continuous and piecewise C^1, and delay $\delta_{v,g}$ is continuous and piecewise C^1.

Remark 5.
If $n_{v,p}(t) = 0$*, then* $D_p^v(t) = A_p^v(t) \implies \delta_{v,p}(t) = 0$*.*
If $n_{v,p}(t) > 0$*, then* $D_p^v(t) < A_p^v(t) \implies [D_p^v]^{-1}\left(A_p^v(t)\right) > t$ *and* $\delta_{v,p}(t) > 0$*.*
Therefore,

$$\forall t, \delta_{v,p}(t) > 0 \Leftrightarrow n_{v,p}(t) > 0 \tag{12}$$

Requirement 2. *First-in-first-out (FIFO) property*
The model satisfies the FIFO property. The delay encountered in queue v *at time* t *is identical for all paths* p *in* P_v*.*

$$\delta_v(t) = \delta_{v,p}(t) = [D_p^v]^{-1}\left(D_p^{\pi_v}(t)\right) - t \qquad \forall t, \forall p \in P_v \tag{13}$$

FIFO property implies that agents exit the queue in the same order that they enter the queue regardless of which path they belong to.

$$t_1 < t_2 \Leftrightarrow [D_{p_1}^v]^{-1}\left(D_{p_1}^{\pi_v}(t_1)\right) < [D_{p_2}^v]^{-1}\left(D_{p_2}^{\pi_v}(t_2)\right) \tag{14}$$

Interpreting A_p^v (resp D_p^v) as the identifier of the agent which arrives in (resp. leaves) queue v at time t, we can see that the queues respect the FIFO rule for each path p. Let x_1 and x_2 be two agents: agent x_1 enters queue v at time t_1^{in} such that $A_p^v\left(t_1^{in}\right) = x_1$ and leaves queue v at time t_1^{out} such that $D_p^v\left(t_1^{out}\right) = x_1$, agent x_2 entered in queue v at time t_2^{in} such that $A_p^v\left(t_2^{in}\right) = x_2$ and leaves queue v at time t_2^{out} such that $D_p^v\left(t_2^{out}\right) = x_2$. As A_p^v and D_p^v are both strictly increasing functions, $t_1^{in} \le t_2^{in} \Rightarrow x_1 \le x_2 \Rightarrow t_1^{out} \le t_2^{out}$, which means that if x_1 is enters queue v before x_2, it will leave v before x_2.

Proposition 1. *FIFO implies conservation of the ratio of flows*
If p_1 *and* p_2 *are two paths in* P_v *such that* $\lambda_{p_1}^{\pi_v}, \lambda_{p_2}^{\pi_v} > 0$*, then the ratio of their flows is conserved when exiting node* v

$$\frac{\lambda_{p_1}^v(t+\delta_v(t))}{\lambda_{p_2}^v(t+\delta_v(t))} = \frac{\lambda_{p_1}^{\pi_v}(t)}{\lambda_{p_2}^{\pi_v}(t)}, \quad \forall t \in (t_{init}, t_{final}] \tag{15}$$

Proof: Let t be an arbitrary time. The FIFO assumption gives $\delta_{v,p}(t) = \delta_v(t)$. By definition of delay $\delta_{v,p}(t)$,

$$D_p^{\pi_v}(t) \quad = \quad D_p^v(t + \delta_{v,p}(t)) \qquad \forall p \in P_v$$

Taking the derivative with respect to t and using $\delta_{v,p}(t) = \delta_v(t)$,

$$\frac{dD_p^{\pi_v}(t)}{dt} \quad = \quad \left(1 + \frac{d\delta_v(t)}{dt}\right) \cdot \left.\frac{dD_p^v}{dt}\right|_{t+\delta_v(t)}$$

Using equation (7) we obtain,

$$\lambda_p^{\pi_v}(t) \quad = \quad \left(1 + \frac{d\delta_v(t)}{dt}\right) \cdot \lambda_p^v(t + \delta_v(t)) \qquad \forall p \in P_v$$

Therefore, it follows that

$$\frac{\lambda_{p_1}^v(t+\delta_v(t))}{\lambda_{p_2}^v(t+\delta_v(t))} \quad = \quad \frac{\lambda_{p_1}^{\pi_v}(t)}{\lambda_{p_2}^{\pi_v}(t)}$$

□

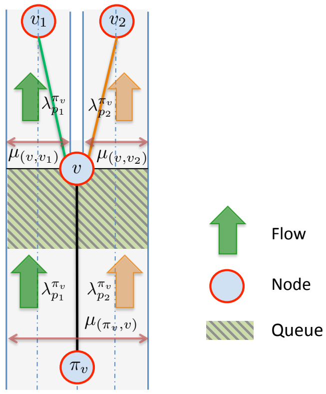

Fig. 1. Diverge model

Definition 4. *Queue state η_v - state transitions*
We define queue state as the boolean valued function $\eta_v(t)$:

$$\eta_v(t) = \begin{cases} 1 & \text{if } \delta_v(t) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

If $\eta_v = 1$, queue v is said to be active, or in active state
If $\eta_v = 0$, queue v is said to be inactive, or in inactive state

A queue state transition happens at time t if

$$\exists \epsilon > 0 \text{ s.t. } \forall \theta \in [-\epsilon, \epsilon], \quad \eta_v(t-\theta) = 1 - \eta_v(t+\theta) \tag{17}$$

When queue v is inactive, $D^v = D^{\pi_v}$.

Definition 5. *Link constraint $c_{v,l}(t)$*
Let $v \in V \backslash \{v_0 \cup S\}$ be a node which is not a source or a sink. For all links $l \in L_v^{out}$, we define the link constraint $c_{v,l}(t)$ as the ratio of arriving flows at time t on capacity at queue v when this flow leaves queue v[4].

$$c_{v,l}(t) = \frac{\sum_{p \in P_l} \lambda_p^{\pi_v}(t)}{\mu_l(t + \delta_v(t))} \tag{18}$$

Definition 6. *Active link $\gamma_v(t)$ and set of active paths $\Gamma_v(t)$ of a node*
We define the active link $\gamma_v(t)$ of a node v at time t as the most constrained link[5] *in L_v^{out}:*

$$\gamma_v(t) \in \arg\max_{l \in L_v^{out}} c_{v,l}(t) \tag{19}$$

We define the set of active paths $\Gamma_v(t)$ in queue v as the set of paths in the most constrained link $\gamma_v(t)$

$$\Gamma_v(t) = P_{\gamma_v(t)} \tag{20}$$

Remark 1 gives $\Gamma_v \subset P_v$.

Requirement 3. *Full capacity discharge property*
The model satisfies the full capacity discharge property. For each node v and time t, if queue v is active at t, then the active link $\gamma_v(t)$ discharges at full capacity.

$$\delta_v(t) > 0 \Rightarrow \sum_{p \in \Gamma_v(t)} \lambda_p^v(t + \delta_v(t)) = \mu_{\gamma_v(t)}(t + \delta_v(t)) \tag{21}$$

With this last property, we complete the definition of the dynamics model.

Definition 7. ***Feasible flows***
A feasible flow λ_p^v at a node v is a flow that satisfies the FIFO, capacity constraint and full capacity discharge properties from requirements 1, 2 and 3.

The definition of the initial conditions on the network completes the definition of the model.

Definition 8. *Initial times for each non-source node*
Given a set of initial delays at each node $\delta_v(t_{initial}) \geq 0, \forall v \in V \backslash (S \cup \{v_0\})$ and an initial time $t_{initial}$, we define the set of initial times over which the departure curves are defined for each non-source node recursively as follows:

$$\begin{cases} t_{0,initial} & = & t_{initial} & \text{for node } v_0 \\ t_{v,initial} & = & t_{\pi_v,initial} + \delta_v(t_{\pi_v,initial}) & \end{cases} \tag{22}$$

D. Existence and uniqueness of the solution to the model

Now that we have fully defined the model dynamics, we consider the well-posedness of the model. In other words, given a network, link capacities and the departure functions at the source, we want to know whether the dynamics of the model admits a unique solution.

Problem 1: General network problem
Input. An arborescence (V, L) with source v_0 and sink set S, capacities $\mu_l(t), \forall l \in L,\ t \in [t_{\text{initial}}, t_{\text{final}}]$, acceptable departure functions from the source $D_p^{v_0} \in \mathbf{D}(t_{\text{initial}}, t_{\text{final}})\ \forall p \in P_{v_0}$ and initial delays $\delta_v(t_{\text{initial}}) \geq 0,\ \forall v \in V \backslash (S \cup \{v_0\})$
Question. Does a corresponding set of feasible flows exist for all internal nodes $v \in V \backslash v_0$ and are they unique?

Theorem 1 stats that the solution to problem 1 both exists and that the solution is unique, under certain conditions on the departure curves at the origin and the link capacities of the network.

Theorem 1. ***Existence and uniqueness of the solution to problem 1***
Problem 1 admits a unique solution under the following conditions.
1) the path flows at the origin $\lambda_p^0(t)$ are piecewise polynomial,
2) link capacities μ_l are piecewise constant over time.

Note that neither of the assumptions of the theorem are restrictive in a practical sense[6].

[4]The dissipation rate of the point queue at the node is only governed by the capacities of the outgoing links. This model can be extended to also impose a discharge rate constraint based on the capacity of the incoming link, but increases the complexity of the notation and the proofs.

[5]When there is a tie, one of them is chosen arbitrarily.

[6]Neither of these assumptions are restrictive in a practical sense, because any piecewise continuous function on a closed interval can be approximated to an arbitrary accuracy by a polynomial of appropriate degree (Stone-Weierstrass theorem [18]) and link capacities do not evolve in a continuous manner. Link capacities are typically subject to discrete changes due to incidents such as accidents and changes in weather.

The next section is devoted to a constructive proof of theorem 1. The general flow of the proof is as follows. Sections III-A-III-C first develop a set of differential equations for delays in the network. In section III-D, we then prove that a unique solution to differential equation on delays also implies a unique solution to problem 1. Section III-E proves that the differential equations on the delay at each node always admit an unique solution, which finally leads to the proof of theorem 1.

III. A SOLUTION BASED ON TIME MAPPING

This section builds a constructive proof of theorem 1. Throughout sections III-A-III-C, we require that the flows at the origin are acceptable departure curves as defined in definition 1 and that the outflows at each node satisfy the model requirements (i.e. result in feasible flows as defined in definition 7).

A. Local study of point queues

We begin by proving proposition 2, which gives an analytical expression for the derivative of the delay at node as a function of its downstream capacities and outgoing flow at its parent nodes.

Proposition 2. *Evolution law of a single queue*
If queue v is active at time t,

$$\left.\frac{d\delta_v}{dt}\right|_t = \frac{\sum_{p\in\Gamma_v(t)} \lambda_p^{\pi_v}(t)}{\mu_{\gamma_v(t)}(t+\delta_v(t))} - 1 \tag{23}$$

The proof of this proposition requires the following lemma.

Lemma 3. *Derivative of queue's length n_v with respect to time*
If node v is active at time t (i.e. t: $\gamma_v(t) = 1$),

$$\sum_{p\in\Gamma_v(t)} \left.\frac{dn_{v,p}}{dt}\right|_{t+\delta_v(t)} = \left[\sum_{p\in\Gamma_v(t)} \lambda_p^{\pi_v}(t+\delta_v(t))\right] - \mu_{\gamma_v(t)}(t+\delta_v(t)) \tag{24}$$

Proof: By definition 2, $n_{v,p}(t) = D_p^{\pi_v}(t) - D_p^v(t)$. Thus,

$$\begin{aligned}\sum_{p\in\Gamma_v} \left.\frac{dn_{v,p}}{dt}\right|_t &= \sum_{p\in\Gamma_v} \left.\left(\frac{dD_p^{\pi_v}}{dt} - \frac{dD_p^v}{dt}\right)\right|_t \\ &= \sum_{p\in\Gamma_v} \left.\left(\lambda_p^{\pi_v} - \lambda_p^v\right)\right|_t\end{aligned} \tag{25}$$

As queue v is active at time t, requirement 3 gives $\sum_{p\in\Gamma_v} \lambda_p^v(t+\delta_{v,t}) = \mu_{\gamma_v(t)}(t+\delta_v(t))$, thus we have

$$\sum_{p\in\Gamma_v(t)} \left.\frac{dn_{v,p}}{dt}\right|_{t+\delta_v(t)} = \left(\sum_{p\in\Gamma_v(t)} \lambda_p^{\pi_v}(t+\delta_{v,t})\right) - \mu_{\gamma_v(t)}(t+\delta_v(t)) \tag{26}$$

□

Lemma 4. *Discharge relationship between queue length and delay*

$$n_{v,p}(t+\delta_v(t)) = D_p^{\pi_v}(t+\delta_v(t)) - D_p^{\pi_v}(t), \quad \forall v \in V, p \in P_v \tag{27}$$

Proof: By definition 2 on queue length, we have $n_{v,p}(t) = D_p^{\pi_v}(t) - D_p^v(t)$, which evaluated at time $t + \delta_{v,p}(t)$ gives $n_{v,p}(t+\delta_{v,p}(t)) = D_p^{\pi_v}(t+\delta_{v,p}(t)) - D_p^v(t+\delta_{v,p}(t))$. From definition 3 on queue delay, we have $D_p^v(t+\delta_{v,p}(t)) = D_p^{\pi_v}(t)$. Combining these two results we obtain,

$$n_{v,p}(t+\delta_v(t)) = D_p^{\pi_v}(t+\delta(t)) - D_p^{\pi_v}(t) \tag{28}$$

□

We can now prove proposition 2.
Proof of proposition 2: Let t be a time such that $\eta_v(t) = 1$. Equation (24) multiplied by $\left(1 + \left.\frac{d\delta_v}{dt}\right|_t\right)$ gives

$$\left(1 + \left.\frac{d\delta_v}{dt}\right|_t\right) \cdot \sum_{p\in\Gamma_v(t)} \left.\frac{dn_{v,p}}{dt}\right|_{t+\delta_v(t)} = \tag{29}$$

$$\left[\left(1 + \left.\frac{d\delta_v}{dt}\right|_t\right) \sum_{p\in\Gamma_v(t)} \lambda_p^{\pi_v}(t+\delta_v(t))\right] - \left(1 + \left.\frac{d\delta_v}{dt}\right|_t\right) \mu_{\gamma_v(t)}(t+\delta_v(t)) \tag{30}$$

Taking the derivative of equation (27) with respect to time and summing over $p \in \Gamma_v$, gives the following equality

$$\left(1 + \left.\frac{d\delta_v}{dt}\right|_t\right) \cdot \sum_{p\in\Gamma_v(t)} \left.\frac{dn_{v,p}}{dt}\right|_{t+\delta_v(t)} = \left[\left(1 + \left.\frac{d\delta_v}{dt}\right|_t\right) \cdot \sum_{p\in\Gamma_v(t)} \lambda_p^{\pi_v}(t+\delta_v(t))\right] - \sum_{p\in\Gamma_v(t)} \lambda_p^{\pi_v}(t) \tag{31}$$

Given equations (29) and (31) have the same left hand side, equalizing their respective right hand sides and simplifying $\left[\left(1 + \left.\frac{d\delta_v}{dt}\right|_t\right) \cdot \sum_{p\in\Gamma_v(t)} \lambda_p^{\pi_v}(t+\delta_{v,p}(t))\right]$ gives the following equation:

$$\left(1 + \left.\frac{d\delta_v}{dt}\right|_t\right) \cdot \mu_{\gamma_v(t)}(t+\delta_v(t)) = \sum_{p\in\Gamma_v(t)} \lambda_p^{\pi_v}(t) \tag{32}$$

Which gives the result,

$$\left.\frac{d\delta_v}{dt}\right|_t = \frac{\sum_{p\in\Gamma_v(t)} \lambda_p^{\pi_v}(t)}{\mu_{\gamma_v(t)}(t+\delta_v(t))} - 1 \tag{33}$$

□

B. Time mapping

The evolution law stated above for any given node v depends on the outgoing flows $\lambda_p^{\pi_v}$ at the parent node. However, this it not an input of Problem 1. In this section, we introduce the notion of time mapping to obtain a modified law for the delay evolution that replaces the outgoing flows at the parent node with the outgoing flow at the origin.

1) Definition of time mapping functions: The evolution law from proposition 2 gives a non-linear *ordinary differential equation* (ODE) that governs the evolution of $\delta_v(t)$. The evolution of delay encountered by an agent x entering queue v at time t depends on the flows entering the queue at t and the capacity of the active link(s) γ_v at time $t + \delta_v(t)$ when agent x leaves the queue. The non-linearity of the ODE makes directly computing the dynamics along a path algebraically complex. Therefore, we introduce a time mapping function.

Let v be an internal node of the network and its parent node be π_v. an agent leaving node π_v at time t will leave node v at time $t+\delta_v(t)$. We now introduce the following time mapping function:

Definition 9. *Node time mapping function T^{v,π_v}*
We define the time mapping function T^{v,π_v} by

$$T^{v,\pi_v} : t \mapsto t+\delta_v(t) \tag{34}$$

an agent leaving node π_v at time t will leave node v at time $T^{v,\pi_v}(t)$

The notation T^{v,π_v} (variable ordering) is chosen for mathematical convenience with respect to the derivatives of the function, as will be apparent in the rest of the discussion. In equation (34), T^{v,π_v} takes a time with a physical meaning at the exit of node π_v on its right hand side, and gives back a time with a physical meaning at the exit of node v on its left hand side.

Proposition 5. *T^{v,π_v} is strictly increasing and bijective*
The function T^{v,π_v} is strictly increasing and thus bijective from its domain to its image. Its derivative is

$$\left.\frac{dT^{v,\pi_v}}{dt}\right|_t = \frac{\sum_{p\in\Gamma_v(t)} \lambda_p^{\pi_v}(t)}{\mu_l\left(t+\delta_v(t)\right)} > 0 \tag{35}$$

Physically, this means that the FIFO assumption is respected: i.e. an agent x_2 entering queue v after another agent x_1 will also leave the queue after x_1

Proof: Taking the derivative of equation (34) and applying equation (23) in proposition 2 gives,

$$\left.\frac{dT^{v,\pi_v}}{dt}\right|_t = \frac{\sum_{p\in\Gamma_v(t)} \lambda_p^{\pi_v}(t)}{\mu_l\left(t+\delta_v(t)\right)}. \tag{36}$$

The departure curves at the origin are strictly increasing since they must be acceptable departure curves. The full capacity discharge property from requirement 3 requires that one outgoing link at each node discharges at full capacity. Finally, these properties combined with proposition 1, which states that the out flows at a node are proportional to the inflows, give us the result that $\left.\frac{dT^{v,\pi_v}}{dt}\right|_t > 0$. □

Thus T^{v,π_v} is invertible and its inverse is an increasing function[7].

Definition 10. *Node time mapping function $T^{\pi_v,v}$ Given an internal node v, we define the function $T^{\pi_v,v}$ as the inverse of T^{v,π_v}*

$$T^{\pi_v,v}\circ T^{v,\pi_v} = \mathbb{1} \text{ and } T^{v,\pi_v}\circ T^{\pi_v,v} = \mathbb{1} \tag{37}$$

We now consider the unique path $(v_0, v_1, \cdots, v_{n-1}, v_n)$ which leads from the source v_0 to some node v_n. As each node has a unique parent, we can recursively trace the path from node v back to the source node v_0. Let t^{v_n} be a fixed time. If an agent x leaves node v_n at the time t^{v_n}, we can recursively define the following:
1) $t^{v_{n-1}}=T^{v_{n-1},v_n}(t^{v_n})$ is the time that agent x left v_{n-1}, $t^{v_n}=t^{v_{n-1}}+\delta_v(t^{v_{n-1}})$
2) $t^{v_{n-2}}=T^{v_{n-2},v_{n-1}}(t^{v_{n-1}})$ is the time that agent x left v_{n-2}, $t^{v_n}=t^{v_{n-2}}+\delta_{v_{n-1}}(t^{v_{n-2}})+\delta_v(t^{v_{n-2}}+\delta_{v_{n-1}}(t^{v_{n-2}}))$ 3) $t^{v_{n-3}}=T^{v_{n-3},v_{n-2}}(t^{v_{n-2}})$ is the time that agent x left v_{n-3}, . . .

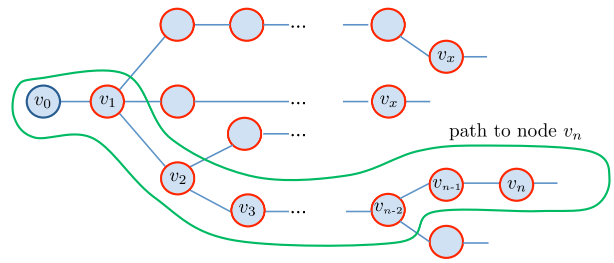

Fig. 2. Time mapping nodes

As T^{v,π_v} and $T^{\pi_v,v}$ are bijective for all internal nodes v, we can give the following definition

Definition 11. *Time mapping function from and to the origin T^{v,v_0} and $T^{v_0,v}$*
Let v_n be a node, and $(v_0, v_1, v_2, \cdots, v_{\pi_n}, v_n)$ be a path from the origin v_0 to node v. We define the time mapping function to the origin as the composition of the node time mapping function on the path between the source and v_n

$$T^{v_0,v_n} = T^{v_0,v_1}\circ T^{v_1,v_2}\circ\cdots\circ T^{v_{\pi_n},v_n} \tag{38}$$

an agent that leaves node v_n at time t left the origin v_0 at time $T^{v_0,v_n}(t)$.

$$T^{v_n,v_0} = T^{v_n,v_{\pi_n}}\circ\cdots\circ T^{v_2,v_1}\circ T^{v_1,v_0} \tag{39}$$

an agent that leaves the origin at time t will leave node v_n at time $T^{v_n,v_0}(t)$

A sample path from the origin v_0 to a node v_n is illustrated in figure 2. We can now define the time mapping function between any arbitrary pair of nodes.

Definition 12. *Time mapping function between two arbitrary nodes*
We define the time mapping function $T^{i,j}$ between node i and node j as follows.
1) There exists a path between nodes i and j (for example nodes v_2 and v_n in figure 2),

$$T^{i,j} = \begin{cases} T^{i,i+1}\circ T^{i+1,i+2}\circ\cdots\circ T^{j-2,j-1}\circ T^{j-1,j} & \text{if } i\prec j \\ T^{i,i-1}\circ T^{i-1,i-2}\circ\cdots\circ T^{j+2,j+1}\circ T^{j+1,j} & \text{if } i\succ j \end{cases} \tag{40}$$

Let x be an agent that leaves node j at time t. $T^{i,j}(t)$ is the time that agent x leaves node j.

2) There does not exist a path between nodes i and j (for example nodes v_2 and v_x in figure 2),

$$T^{i,j} = T^{i,v_0}\circ T^{v_0,j} \tag{41}$$

Let x_j be an agent that leaves node j at t. From definition 11 we know that x_j leaves the origin at time $T^{0,j}(t)$. Let x_i be an agent that also leaves the origin at time $T^{0,j}(t)$. Then $T^{i,j}(t)$ is the time that agent x_i leaves node i.

[7] If the acceptable set of departure curves $\mathbf{D}$ is relaxed to allow monotonically increasing instead of strictly increasing functions, T^{v,π_v} becomes a correspondence, and the mathematical treatment would be more involved.

Definition 13. *Time mapping operator* $\mathbf{T}^{i,j}$
We define the time mapping operator $\mathbf{T}^{i,j}$ *on the set* F *of time dependent functions as follows:*

$$\begin{array}{rccc} \mathbf{T}^{i,j}: & F & \rightarrow & F \\ & f & \mapsto & f \circ T^{j,i} \end{array} \tag{42}$$

We now consider the physical interpretation of $T^{i,j}$.

2) Time mapping of model quantities: This section first studies the relationship between departure curves at different nodes and the time mapping function. We then define the time mapped versions of the other quantities in the model. The time mapping operators allow for mapping any quantity from one node to the other. This definition of a time mapped quantities thus allows any quantity to be defined with respect to the source node of the network.

Proposition 6. *Physical interpretation of the time mapping function*
Let p *be a path, and* $(v_0, v_1, v_2, \cdots, v_n)$ *be a sequence of consecutive nodes on the path.*

$$D_p^{v_i} = D_p^{v_0} \circ T^{v_0,v_i} \qquad \forall v_i \in p \tag{43}$$

Let $x = D_p^{v_0}(t^{v_0})$ *be an agent on path* p *that leaves the origin at time* t^{v_0} *and* $t^{v_i} = T^{v_i,0}(t^{v_0}) \forall v_i \in p$.

$$D_p^{v_0}(t^{v_0}) = D_p^{v_1}(t^{v_1}) = \cdots = D_p^{v_i}(t^{v_i}) = \cdots = D_p^{v_n}(t^{v_n}) \tag{44}$$

Proof: Proof by induction on the length of the sequence k. If $k = 0$, the result is trivial. Let $k \in [1, i]$ be an integer. By the induction hypothesis, we assume that the result is true for to $k = i-1$, i.e. $D_p^{v_{i-1}} = D_p^{v_0} \circ T^{0,v_{i-1}}$. By the definition of path delay $\delta_{v,p}, D_p^{v_i}(t + \delta_{v_i,p}(t)) = D_p^{v_{i-1}}(t), \forall t$, which means $D_p^{v_{i-1}} = D_p^{v_i} \circ T^{v_i,v_{i-1}}$. Composing both sides of the equality with T^{v_{i-1},v_i} we get $D_p^{v_i} = D_p^{v_{i-1}} \circ T^{v_{i-1},v_i}$. Substituting the induction hypothesis and simplifying the results completes the proof.

$$\begin{aligned} D_p^{v_i} &= D_p^{v_{i-1}} \circ T^{v_{i-1},v_i} \\ &= D_p^{v_0} \circ T^{0,v_{i-1}} \circ T^{v_{i-1},v_i} \\ &= D_p^{v_0} \circ T^{v_0,v_i} \end{aligned}$$

Equation (44) follows directly from equation (43). □

Remark 6. *As function* $T^{i,j}$ *is the inverse of* $T^{j,i}$*, the operator* $\mathbf{T}^{j,i}$ *is the inverse of* $\mathbf{T}^{i,j}$.

We can now reformulate the first equation of proposition 6 as follows:

Proposition 7. *Time mapping of departure curve* D_p^v
Let i *and* j *be two nodes on path* p.

$$D_p^i = \mathbf{T}^{i,j}(D_p^j) \tag{45}$$

Proof: Using definition 13 we have,

$$\begin{aligned} \mathbf{T}^{i,j}\left(D_p^j\right) &= D_p^j \circ T^{j,i} = D_p^j \circ T^{j,0} \circ T^{0,i} \\ &= D_p^0 \circ T^{0,i} = D_p^i \end{aligned}$$

□

Proposition 8. *Time mapping and flows*
Let v *be a node on path* p.

$$\lambda_p^i = \mathbf{T}^{i,j}\left(\lambda_p^j\right) \cdot \frac{dT^{j,i}}{dt} \tag{46}$$

Proof: From the definition of flow, $\lambda_p^v = \dfrac{dD_p^v}{dt}$. The result is obtained by simply taking the derivative of the equation $D_p^i = D_p^j \circ T^{j,i}$ (from proposition 7) with respect to time. □

Remark 7. *The time mapping and derivative operators do not commute.*

Definition 14. *Time mapping of delay* δ_j^i
Let v *be an internal node*[8]*. We define the time mapped delay in queue* v *at node* π_v, $\delta_v^{\pi_v}$ *as the delay encountered in queue* v *by an agent leaving node* π_v*:*

$$\delta_v^{\pi_v} \doteq \delta_v \tag{47}$$

Let i *be an arbitrary node and* j *be an internal node. We define the time mapped delay in queue* j *at node* i, δ_j^i *as*

$$\delta_j^i \doteq \mathbf{T}^{i,\pi_j}\left(\delta_j^{\pi_j}\right) = \delta_j^{\pi_j} \circ T^{\pi_j,i} \tag{48}$$

Physically, if nodes i *and* j *are on the same branch with* $i \prec j$ *(resp.* $i \succ j$*), then* $\delta_j^i(t)$ *is the time that an agent which leaves queue* i *at time* t *will be (resp. has been) delayed at in queue* j.

Definition 15. *Time mapping for capacity*
We define the time mapped capacity of a link l, $\mu_l^{v_l^{in}}$ *as the capacity encountered by an agent at queue* v_l^{in} *in link* l

$$\mu_l^{v_l^{in}} \doteq \mu_l \tag{49}$$

Let l *be an arbitrary link and* v *an internal node. We define the time mapped capacity of link* l *at node* v *as*

$$\mu_l^v \doteq \mathbf{T}^{v,v_l^{in}}\left(\mu_l^{v_l^{in}}\right) = \mu_l^{v_l^{in}} \circ T^{v_l^{in},v} \tag{50}$$

Physically, if link l *and node* v *are on the same branch with* $v_l^{in} \prec v$ *(resp.* $v_l^{in} \succ v$*), then* $\mu_v^l(t)$ *is the capacity an agent that leaves queue* v *at time* t *encountered (resp. encounters) at link* l.

Proposition 9. *Physical interpretation of mapped delay and mapped capacity*
Let v_j *be an arbitrary node,* p *be a path, and* $(v_0, v_1, v_2, \cdots, v_n)$ *be a sequence of consecutive nodes on the path* p*. Also, let* $t^{v_i} = T^{v_i,0}(t^{v_0}), \forall v_i \in p$.

$$\delta_{v_j}^{v_0}(t^{v_0}) = \delta_{v_j}^{v_1}(t^{v_1}) = \cdots = \delta_{v_j}^{v_i}(t^{v_i}) = \cdots = \delta_{v_j}^{v_n}(t^{v_n}) \tag{51}$$

Let l *be an arbitrary link.*

$$\mu_l^{v_0}(t^{v_0}) = \mu_l^{v_1}(t^{v_1}) = \cdots = \mu_l^{v_i}(t^{v_i}) = \cdots = \mu_l^{v_n}(t^{v_n}) \tag{52}$$

Proof: Let i be an arbitrary node and j be an internal node. From definition (14) for time mapped delay we have.

$$\begin{aligned} \delta_j^i\left(t^i\right) &\doteq \delta_j^{j-1}\left(T^{j-1,i}\left(t^i\right)\right) \\ &= \delta_j^{j-1}\left(t^{j-1}\right) \end{aligned}$$

[8]An internal node is a node v which is neither a sink nor the source $k \in K \backslash (\{0\} \cup S)$

Therefore, $\delta_{v_j}^{v_i}\left(t^{v_i}\right)=\delta_{v_j}^{v_{j-1}}\left(t^{v_{j-1}}\right), \forall v_i \in p$, which proves equation (51). The proof for equation (52) is identical. □

Definition 16. *Time mapping of active link and active paths*
Let v be an internal node. We define mapped active link $\gamma_v^{\pi_v}$ as the active link for flow exiting node π_v at queue v, and mapped active paths $\Gamma_v^{\pi_v}$ as the active paths for flow exiting node π_v at queue v.

$$\gamma_v^{\pi_v} \doteq \gamma_v \quad ; \quad \Gamma_v^{\pi_v} \doteq \Gamma_v \tag{53}$$

Let j be an arbitrary node, we define the mapped active link and mapped paths for flow exiting queue v at node j as

$$\gamma_v^j=\mathbf{T}^{j,\pi_v}\left(\gamma_v^{\pi_v}\right) \quad ; \quad \Gamma_j^v=\mathbf{T}^{j,\pi_v}\left(\Gamma_v^{\pi_v}\right) \tag{54}$$

Physically, if node j and node v are on the same branch with $j \prec v$ (resp. $j \succ v$), then $\gamma_v^j(t)$ is the active link that an agent leaving node j at time t will encounter (resp. encountered) at queue v, and $\Gamma_j^v(t)$ are the corresponding active paths.

Definition 17. *Time mapped link constraint*
Let v be a internal node and $l \in L_v^{out}$. We define the mapped link constraint $c_{v,l}^{\pi_v}$ as the link constraint at link l for an agent leaving node π_v.

$$\begin{aligned} c_{v,l}^{\pi_v}(t) &\doteq \frac{\sum_{p\in P_l}\lambda_p^{\pi_v}(t)}{\mu_l\left(t+\delta_v(t)\right)} && (55)\\ &= \frac{\sum_{p\in P_l}\lambda_p^{\pi_v}(t)}{\mu_l^v\left(t+\delta_v(t)\right)} \\ &= \frac{\sum_{p\in P_l}\lambda_p^{\pi_v}(t)}{\mu_l^{\pi_v}(t)} && (56)\end{aligned}$$

Let j be an arbitrary node, we define the mapped link constraint for link l at node j as

$$c_{v,l}^j \doteq \mathbf{T}^{j,\pi_v}\left(c_{v,l}^{\pi_v}\right)=c_{v,l}^{\pi_v}\circ T^{\pi_v,j} \tag{57}$$

$$c_{v,l}^j(t)=\frac{\sum_{p\in P_l}\lambda_p^j(t)}{\mu_l^j(t)}\cdot\frac{dT^{\pi_v,j}}{dt} \tag{58}$$

Physically, if node j and node v are on the same branch with $j \prec v$ (resp. $j \succ v$), then $c_{v,l}^j(t)$ is the link constraint that an agent leaving node j at time t will encounter (resp. encountered) at link l.

Remark 8. *The notation of the link constraint can be simplified for convenience as follows when time mapped.*

$$c_{v,l}^j=c_l^j \tag{59}$$

We use the simplified notation in the rest of the discussion.

Proposition 10. *The mapping of link constraints and active links is coherent*
For all non-sink nodes $j \in V\backslash S$, internal nodes $v \in V\backslash(S \cup \{0\}$ and time $t \in (t_{initial}, t_{final}]$, we have

$$\gamma_v^j(t) \in \underset{l\in L_v^{out}}{\arg\max}\, c_l^j(t) \tag{60}$$

Proof: Let v be an internal node and let t^j be a time. Let $t^{\pi_v}=T^{\pi_v,j}\left(t^j\right)$. Proving the proposition is equivalent to proving the following set equality

$$\underset{l\in L_v^{\text{out}}}{\arg\max}\, c_{v,l}\left(t^{\pi_v}\right)=\underset{l\in L_v^{\text{out}}}{\arg\max}\, c_l^j\left(t^j\right) \tag{61}$$

From the definition of the link constraint in equation (18) we have

$$c_{v,l}\left(t^{\pi_v}\right) \doteq \frac{\sum_{p\in P_l}\lambda_p^{\pi_v}\left(t^{\pi_v}\right)}{\mu_l\left(t^{\pi_v}+\delta_v\left(t^{\pi_v}\right)\right)} \tag{62}$$

By definition of μ_l^v in equation (49), we have $\mu_l\left(t^{\pi_v}+\delta_v\left(t^{\pi_v}\right)\right)=\mu_l^v\left(t^{\pi_v}+\delta_v\left(t^{\pi_v}\right)\right)$ and defining $t^v \doteq T^{v,\pi_v}\left(t^{\pi_v}\right)=t^{\pi_v}+\delta_v\left(t^{\pi_v}\right)$, we obtain $\mu_l\left(t^{\pi_v}+\delta_v\left(t^{\pi_v}\right)\right)=\mu_l^v\left(T^{v,\pi_v}\left(t^{\pi_v}\right)\right)=\mu_l^v\left(t^v\right)$. Equation (52) finally gives

$$\mu_l\left(t^{\pi_v}+\delta_v\left(t^{\pi_v}\right)\right)=\mu_l^j\left(t^j\right) \tag{63}$$

Moreover, using equation (46) gives $\lambda_p^{\pi_v}\left(t^{\pi_v}\right)\cdot\left.\frac{dT^{\pi_v,j}}{dt}\right|_{t^j}=\lambda_p^j\left(t^j\right)$. Summing on all paths p in P_l, we obtain

$$\sum_{p\in P_l}\lambda_p^{\pi_v}\left(t^{\pi_v}\right)=\frac{1}{\left.\frac{dT^{\pi_v,j}}{dt}\right|_{t^j}}\cdot\sum_{p\in P_l}\lambda_p^j\left(t^j\right) \tag{64}$$

Substituting equations (63) and (64) in the right hand side of equation (62) and using the time mapped link constraint from equation (58), we obtain

$$\begin{aligned} c_{v,l}\left(t^{\pi_v}\right) &= \frac{1}{\left.\frac{dT^{\pi_v,j}}{dt}\right|_{t^j}}\cdot\left[\frac{\sum_{p\in P_l}\lambda_p^j\left(t^j\right)}{\mu_l^j\left(t^j\right)}\right] && (65)\\ &= \frac{1}{\left.\frac{dT^{\pi_v,j}}{dt}\right|_{t^j}}\cdot c_l^j\left(t^j\right) && (66)\end{aligned}$$

For all $l \in L_v^{\text{out}}$, $c_{v,l}\left(t^{\pi_v}\right)$ and $c_l^j\left(t^j\right)$ are proportional (and the proportionality ratio is independent from l). Therefore, the $\arg\max$ in equation (61) are the same. which concludes the proof. □

Definition 18. *Capacity of the active link*
For notational simplicity we denote the capacity of the active link of an agent that enters queue v at time t as follows:

$$\begin{aligned} Q_v(t) &\doteq \mu_{\gamma_v(t)}^{\pi_v}(t) && (67)\\ &= \mu_{\gamma_v(t)}^{v}\left(t+\delta_v(t)\right) \\ &= \mu_{\gamma_v(t)}\left(t+\delta_v(t)\right) && (68)\end{aligned}$$

Definition 19. *Time mapped capacity of the active link*
Let v be a internal node. We define the time mapped active link capacity $Q_v^{\pi_v}$ as the capacity of link γ_v as seen by an agent at node π_v.

$$Q_v^{\pi_v} \doteq Q_v \tag{69}$$

Let j be an arbitrary node, we define the mapped active link capacity for link $\gamma_v(t)$ as seen by an agent at node j as

$$Q_v^j \doteq \mathbf{T}^{j,\pi_v}\left(Q_v^{\pi_v}\right)=Q_v^{\pi_v}\circ T^{\pi_v,j}=Q_v\circ T^{\pi_v,j} \tag{70}$$

Physically, if node j and node v are on the same branch with $j \prec v$ (resp. $j \succ v$), then $Q_v^j(t)$ is the active link capacity that an agent leaving node j at time t will encounter (resp. encountered) at link $\gamma_v^j(t)$.

Definition 20. *Time mapping of queue state*
Let i be an arbitrary node and j be an internal node. We

define the time mapped queue state of queue j at node i, η_j^i as the queue state at queue j as seen by an agent at queue i

$$\eta_j^i \doteq \mathbf{T}^{i,\pi_v}\left(\eta_j^{\pi_v}\right) = \eta_j^{\pi_v} \circ T^{\pi_v,i} = \eta_j \circ T^{\pi_v,i} \tag{71}$$

Physically, if queue i and node j are on the same branch with $i \prec j$ (resp. $i \succ j$), then $\eta_j^i(t)$ is the queue state an agent that leaves node i at time t encounters (resp. encountered) at queue j.

C. Global evolution of delay

We now have the necessary tools to define the evolution of delays at any node of the network with respect to the flows at any upstream node in the network.

Definition 21. *First active upstream node*
Let v be an internal node. We define the first active upstream node of v as

$$\Upsilon_v^j(t) = \max_{\succ}\left\{u | u \prec v, \eta_u^j(t) = 1\right\} \tag{72}$$

For notational convenience we also define the following:

$$\hat{\gamma}_v^j(t) \doteq \gamma^j_{\Upsilon_v^j(t)}(t) \tag{73}$$

$$\hat{\Gamma}_v^j(t) \doteq \Gamma^j_{\Upsilon_v^j(t)}(t) \tag{74}$$

$$\hat{Q}_v^j(t) \doteq Q^j_{\Upsilon_v^j(t)}(t) \tag{75}$$

$$\hat{\eta}_v^j(t) \doteq \eta^j_{\Upsilon_v^j(t)}(t) \tag{76}$$

Theorem 2. ***Evolution law for delay at an arbitrary internal node v mapped to any node j***
Given an arbitrary internal node $v \in V\backslash(S \cup \{0\})$ such that queue v is active, if the flows at the origin are acceptable departure curves and the model requirements are satisfied, the evolution law for delay mapped to any upstream node $j \in V\backslash S$ is

$$\left.\frac{d\delta_v^j}{dt}\right|_t = \begin{cases} \dfrac{\sum_{p\in\Gamma_v^j(t)} \lambda_p^j(t)}{Q_v^j(t)} - \left.\dfrac{dT^{0,j}}{dt}\right|_t & \text{if } v \text{ is the first active queue} \in p \\ \dfrac{\sum_{p\in\Gamma_v^j(t)} \lambda_p^j(t)}{Q_v^j(t)} - \dfrac{\sum_{p\in\hat{\Gamma}_p^j(t)} \lambda_p^j(t)}{\hat{Q}_v^j(t)} & \text{otherwise} \end{cases} \tag{77}$$

Proof: Let t be a time and v be a node. Evolution law (23) in proposition 2 gives

$$\left.\frac{d\delta_v}{dt}\right|_t = \frac{\sum_{p\in\Gamma_v(t)} \lambda_p^{\pi_v}(t)}{Q_v(t)} - 1 \tag{78}$$

By the definition of the time mapping functions we have, $\delta_v^{\pi_v}(t) \doteq \delta_v(t), Q_v^{\pi_v}(t) \doteq Q_v(t), \Gamma_v^{\pi_v}(t) \doteq \Gamma_v(t)$. Thus, equation (78) becomes:

$$\left.\frac{d\delta_v^{\pi_v}}{dt}\right|_t = \frac{\sum_{p\in\Gamma^{\pi_v}_v(t)} \lambda_p^{\pi_v}(t)}{Q_v^{\pi_v}(t)} - 1 \tag{79}$$

Case 1: If node v is not the first active node of path p and $\Upsilon_v(t)$ exists.

Let for an arbitrary node j, $t^j = T^{j,\pi_v}(t)$. Since all the nodes between $\Upsilon_v(t)$ and π_v are inactive by the definition of $\Upsilon_v(t)$, we have

$$t^{\Upsilon_v(t^{\pi_v})} = t^{\pi_v} = t \tag{80}$$

Furthermore, since $\hat{\eta}_v^{\pi_v}(t) = 1$, and the full capacity discharge of active links (assumption 3), we have

$$\sum_{p\in\hat{\Gamma}_v^{\pi_v}(t)} \lambda_p^{\Upsilon_v(t)}(t) = \hat{Q}_v^{\pi_v}(t) \tag{81}$$

$$\sum_{p\in\hat{\Gamma}_v^{\pi_v}(t)} \lambda_p^{\pi_v}(t) = \hat{Q}_v^{\pi_v}(t) \tag{82}$$

Thus:

$$\frac{\sum_{p\in\hat{\Gamma}_v^{\pi_v}(t)} \lambda_p^{\pi_v}(t)}{\hat{Q}_v^{\pi_v}(t)} = 1 \tag{83}$$

By replacing the constant 1 in equation (79) with the above result we get,

$$\left.\frac{d\delta_v^{\pi_v}}{dt}\right|_t = \frac{\sum_{p\in\Gamma_v^{\pi_v}(t)} \lambda_p^{\pi_v}(t)}{Q_v^{\pi_v}(t)} - \frac{\sum_{p\in\hat{\Gamma}_v^{\pi_v}(t)} \lambda_p^{\pi_v}(t)}{\hat{Q}_v^{\pi_v}(t)} \tag{84}$$

This gives us the result for $j = \pi_v$. We will now map this result to any node $j \in V\backslash S$. By definition of time mapping, we have

$$\delta_v^j = \delta_v^{\pi_v} \circ T^{\pi_v,j} \tag{85}$$

Taking its derivative with respect to time, we obtain

$$\frac{d\delta_v^j}{dt} = \left[\frac{d\delta_v^{\pi_v}}{dt} \circ T^{\pi_v,j}\right] \cdot \frac{dT^{\pi_v,j}}{dt} \tag{86}$$

$$\left.\frac{d\delta_v^j}{dt}\right|_t = \left[\frac{\sum_{p\in\Gamma_v^{\pi_v}\circ T^{\pi_v,j}(t)} \lambda_p^{\pi_v} \circ T^{\pi_v,j}(t)}{Q_v^{\pi_v} \circ T^{\pi_v,j}(t)} - \frac{\sum_{p\in\Gamma^{\pi_v}_{\Upsilon_v t}\circ T^{\pi_v,j}(t)} \lambda_p^{\pi_v} \circ T^{\pi_v,j}(t)}{Q^{\pi_v}_{\Upsilon_v(t)} \circ T^{\pi_v,j}(t)}\right] \cdot \left.\frac{dT^{\pi_v,j}}{dt}\right|_t \tag{87}$$

Equation (46) on flow mapping gives

$$\left(\lambda_p^{\pi_v} \circ T^{\pi_v,j}(t)\right) \cdot \left.\frac{dT^{\pi_v,j}}{dt}\right|_t = \lambda_p^j(t) \tag{88}$$

Substituting this result and the simple time mapping transformations of λ and Q into equation (87) gives the final result

$$\left.\frac{d\delta_v^j}{dt}\right|_t = \frac{\sum_{p\in\Gamma_v^j(t)} \lambda_p^j(t)}{Q_v^j(t)} - \frac{\sum_{p\in\Gamma^j_{\Upsilon_v(t)}(t)} \lambda_p^j(t)}{Q^j_{\Upsilon_v(t)}(t)} \tag{89}$$

$$= \frac{\sum_{p\in\Gamma_v^j(t)} \lambda_p^j(t)}{Q_v^j(t)} - \frac{\sum_{p\in\hat{\Gamma}_v^j(t)} \lambda_p^j(t)}{\hat{Q}_v^j(t)} \tag{90}$$

Case 2: If node v is the first active node of path p, we leave the constant 1 in equation (79) and follow the same remaining steps as in case 1 to obtain the result.

$$\left.\frac{d\delta_v^j}{dt}\right|_t = \frac{\sum_{p\in\Gamma_v^j(t)} \lambda_p^j(t)}{Q_v^j(t)} - \left.\frac{dT^{0,j}}{dt}\right|_t \tag{91}$$

□

Applying theorem 2 with $j = 0$, we see that the delays with respect to the flows at the origin δ_v^0 are solutions to the ordinary differential equations in definition 22.

Definition 22. *Time mapped delay evolution differential equation*

- *If v is not an active node and the flow on its active link γ_0^v is within capacity, then* $\dfrac{d\delta_v^0}{dt} = 0$.
- *If v is an active node or its active link γ_0^v is over capacity, then*

$$\left.\frac{d\delta_v^0}{dt}\right|_t = \begin{cases} \dfrac{\sum_{p\in\Gamma_v^0(t)} \lambda_p^0(t)}{Q_v^0(t)} - 1 & \text{if } v \text{ is the first active queue} \in p \\ \dfrac{\sum_{p\in\Gamma_v^0(t)} \lambda_p^0(t)}{Q_v^0(t)} - \dfrac{\sum_{p\in\hat{\Gamma}_v^0(t)} \lambda_p^0(t)}{\hat{Q}_v^0(t)} & \text{otherwise} \end{cases} \tag{92}$$

where the time mapping functions are redefined from delays as follows:

$$T^{j,0} = \sum_{0\prec i \preccurlyeq j} \delta_i^0 \tag{93}$$

Proposition 11. *Delay evolution does not depend on departure curves*
All the time mapped quantities in equations (92) *can be computed using only the initial delays, departure curve at the origin and the link capacities. It does not require the departure curves for any internal nodes $v \in V\backslash v_0$.*

Proof: The time mapping function only depends on the delay functions from definition 9. The time mapped flows can be obtained using the time mapping function using proposition 8. The other time mapped quantities are by definition constructed using the time mapping function as given in section III-B2. □

D. Equivalence of departure curves and delays

We prove Theorem 1 on the existence and uniqueness of Problem 1 by first showing the equivalence between Problem 1 and Problem 2 (defined below), and then proving the existence and uniqueness of Problem 2 in the next section.

Problem 2: General delay problem
Input. An arborescence (V, L) with source v_0 and sink set S, capacities $\mu_l(t), \forall l \in L,\ t \in [t_{\text{initial}}, t_{\text{final}}]$, departure functions from the source $D_p^{v_0} \in \mathbf{D}(t_{\text{initial}}, t_{\text{final}})\ \forall p \in P_{v_0}$ and initial delays $\delta_v(t_{\text{initial}}) \geq 0,\ \forall v \in V\backslash(S \cup \{v_0\})$
Question. Does a solution to the time mapped delay function from definition 22 for each node $v \in V\backslash v_0$ exist and is it unique?

Theorem 3. *Problem (1) and problem (2) are equivalent*

Proof: The inputs to both problems are identical. Therefore, we only need to prove that the existence of a solution to one problem implies a unique and feasible corresponding solution to the other problem.

($\Rightarrow$) Suppose first that Problem 1 admits a solution.
By the definition of delay,

$$\delta_v(t) = [D_p^v]^{-1}\left(D_p^{\pi_v}(t)\right) - t, \tag{94}$$

By the definition of time mapped delay we obtain,

$$\begin{aligned} \delta_v^0(t) &= \delta_v^{\pi_v}(t) \circ T^{\pi_v,0}(t) && (95)\\ &= \delta_v^{\pi_v}\left([D_p^v]^{-1}\left(D_p^{\pi_v}(t)\right) - [D_p^0]^{-1}\left(D_p^{\pi_v}(t)\right)\right) && (96)\\ &= \delta_v\left([D_p^v]^{-1}\left(D_p^{\pi_v}(t)\right) - [D_p^0]^{-1}\left(D_p^{\pi_v}(t)\right)\right), && (97)\end{aligned}$$

which can be made a function of only D_p^v by equation (94).

Theorem 2 then ensures that the delay functions thus defined satisfy the time mapped delay evolution from definition 22, i.e. a feasible solution to problem 2. Furthermore, the solution is unique from equation (97), since D_p^v is a strictly increasing function.

($\Leftarrow$) Suppose now that Problem 2 admits a solution $\delta_v^0(t)$. We can build the corresponding departure curves $D_p^v(t)$ as follows.

$$D_p^0(t) = \delta_0^0(t) \tag{98}$$

The inverse departure curve $[D_p^0]^{-1}(x)$ can be constructed from $D_p^0(t)$, since the departure curve is strictly increasing.

$$[D_p^v]^{-1}(x) = [D_p^0]^{-1}(x) + T^{v,0}\left([D_p^0]^{-1}(x)\right) \tag{99}$$

The departure curve $D_p^v(t)$ can also be constructed from $[D_p^v]^{-1}(x)$ due to the strictly increasing nature of the functions.

We now show that the departure curves thus defined are feasible departure curves, i.e. a feasible solution to problem 1.

1) D_p^0 is continuous and piecewise C_1 because λ_p^0 is piecewise continuous. Furthermore, since $T^{j,0}$ is strictly increasing for all nodes j, D_p^v is continuous and piecewise C_1.
2) The capacity constraint on links is imposed by equation (92) due to proposition 10.
3) The FIFO condition is satisfied by construction since the delay δ_v^0 is not a function of the path p.
4) The full capacity discharge of the active queues is also imposed by by equation (92) due to proposition 10.

■

E. Existence and uniqueness of the time mapped delay evolution

This section proves Theorem 4 on the existence and uniqueness of the solution to Problem 2.

Theorem 4. ***Existence and uniqueness of the solution to problem (2)***
The solution to problem (2) exists and is unique on the time interval of the problem $[t_{initial}, t_{final}]$, if the following conditions are satisfied.

1) *the path flows at the origin $\lambda_p^0(t)$ are piecewise polynomial,*
2) *link capacities μ_l are piecewise constant over time.*

The proof of this theorem is fairly technical and requires several definitions and lemmas. Theorem 1 is a direct corollary of this result due to to Theorem 3 on the equivalence of the two problems.

The main goal of the proof of Theorem 4 is to show that there are a finite number of possible transitions, and to integrate equation (92) across the transitions. The next definitions and lemmas enables to establish these properties.

Definition 23. *Depth of a node $d(v)$*
We define the depth $d(v)$ of a node v as the number of links on the unique path from the origin v_0 to node v

Definition 24. *Link constraint comparators $B_{(c_{l_1},c_{l_2})}(t)$ and $B_{c_l}(t)$*
Given a node v and two distinct links (l_1, l_2), we define the boolean comparator $B_{(c_{l_1},c_{l_2})}(t)$ as follows:

$$B_{(c_{l_1},c_{l_2})}(t) = \begin{cases} 1 & \text{if } \frac{\sum_{p\in P_{l_1}} \lambda_p^0(t)}{\mu_{l_1}^0(t)} > \frac{\sum_{p\in P_{l_2}} \lambda_p^0(t)}{\mu_{l_2}^0(t)} \\ 0 & \text{otherwise} \end{cases} \tag{100}$$

Given a node v and link $l \in L_v^{out}$, we define the boolean comparator $B_{c_l}(t)$ as follows:

$$B_{c_l}(t) = \begin{cases} 1 & \text{if } \frac{\sum_{p\in P_l} \lambda_p^0(t)}{\mu_l^0(t)} > 1 \\ 0 & \text{otherwise} \end{cases} \tag{101}$$

Definition 25. *Time segment of constant link constraint J*
A time segment J is a segment of constant link constraint if and only if

1) *for each each $l \in L$, the boolean $B_{c_l}(t)$ is constant on J,*
2) *for each each pair of nodes $(l_1, l_2) \in L$, the boolean $B_{(c_{l_1},c_{l_2})}(t)$ is constant on J.*
3) *for each each $l \in L$, the time mapped link capacity $\mu_l^0(t)$ is constant on J.*

Lemma 12. *Under the assumptions on flows and capacities, there are a finite number of segments of constant link constraint*

Proof: Consider a pair of links (l_1, l_2). Since capacities are piecewise constant and flows are piecewise polynomial, there are a finite number of segments on which the capacities are constant and flows are polynomial. On any such a segment, $\frac{\sum_{p\in P_{l_1}} \lambda_p^0(t)}{\mu_{l_1}^0(t)}$ and $\frac{\sum_{p\in P_{l_1}} \lambda_p^0(t)}{\mu_{l_1}^0(t)} - \frac{\sum_{p\in P_{l_2}} \lambda_p^0(t)}{\mu_{l_2}^0(t)}$ are polynomials. Therefore, the number of times each expression crosses zero is bounded by the degree of the polynomial, which implies that there are a finite number of segments of constant link constraint. ☐

Lemma 13. *Constant active link*
If J is a segment of constant link constraint, the active link γ_v^0 of any node v is constant on J.

Proof: The result comes directly from the definition of a segment of constant constraint. ☐

Definition 26. *Solution of depth n*
A solution of problem (2) for depth n is a set of solutions δ_v for all nodes v such that $d(v) < n$. It can be rigorously defined because the equations for δ_v only depend on variables associated with nodes of depth less than n.

Definition 27. *Elementary time segment $T^e(v)$*
Given a node v and a solution of depth $d(v) - 1$ (if v is not the origin), an elementary segment for node v is a time segment $T^e(v)$ such that

- *$T^e(v)$ is a segment of constant constraint,*
- *If v is not the origin, for each node $j \in V$ such that $d(j) < d(v)$, the node state $\eta_j(t)$ is constant on $T^e(v)$.*

Lemma 14. *Single transition of node state on an elementary segment*
If there exists a solution to problem (2) up to depth $d(v-1)$, and if $T^e(v) = [t_0, t_f]$ is an elementary segment for node v, then there is a solution δ_v^0 of the problem and node v admits at most one transition in $T^e(v)$.

Proof: As for each node $j \in V$ such that $d(j) < d(v)$, the node state $\eta_j(t)$ is constant on $T^e(v)$, the first active upstream node Υ_v is constant over time. Moreover, as $T^e(v)$ is a segment of constant constraint, Lemma 13 gives that active link γ_v and first active upstream link $\hat{\gamma}_v$ are constant on $T^e(v)$, and the sign of $\frac{\sum_{p\in\Gamma_v^0(t)} \lambda_p^0(t)}{Q_v^0(t)} - \frac{\sum_{p\in\hat{\Gamma}_v^0(t)} \lambda_p^0(t)}{\hat{Q}_v^0(t)}$ is constant on $T^e(v)$.

Let us now consider the following four cases:

1) $B_{(c_{\hat{\gamma}_v},c_{\gamma_v})}(t_0) = 1, \eta_v^0(t) = 1$
$\implies \left.\frac{d\delta_v^0}{dt}\right|_t > 0$ and since the queue state is already active no transition will occur.
2) $B_{(c_{\hat{\gamma}_v},c_{\gamma_v})}(t_0) = 1, \eta_v^0(t) = 0$
$\implies \left.\frac{d\delta_v^0}{dt}\right|_t > 0$ and the queue state will immediately transition to being active $\eta_v^0(t) = 1$. No further transitions will occur as shown above.
3) $B_{(c_{\hat{\gamma}_v},c_{\gamma_v})}(t_0) = 0, \eta_v^0(t) = 1$
$\implies \left.\frac{d\delta_v^0}{dt}\right|_t \leq 0$ and the queue at node v starts dissipating. There will be a transition in the queue state to inactive $\eta_v^0(t) = 0$ if the queue dissipates by time t_f and the queue state will remain active otherwise.
4) $B_{(c_{\hat{\gamma}_v},c_{\gamma_v})}(t_0) = 0, \eta_v^0(t) = 0$
$\implies \left.\frac{d\delta_v^0}{dt}\right|_t \leq 0$ and the only possibility is the strict equality case and the queue state remains inactive.

☐

Lemma 15. *Unique solution on an elementary segment*
Let $T^e(v)$ be an elementary segment for node v. Assuming a solution of depth $d(v) - 1$ (if v is not the origin), then solution of equation (92) for node v exists is unique on $T^e(v)$.

Proof: By Lemma 14, there can be at most one state transition of node v in $T^e(v)$. This splits $T^e(v)$ into at most two sub-segments where $\eta_v = 0$ or $\eta_v = 1$. From Lemma 13 we have that active link γ_v and the first active upstream link Υ_v are constant on $T^e(v)$. Therefore, the quantities $\Gamma_v, \hat{\Gamma}_v, Q_v$ and $\hat{Q}_v$ are constant on $T^e(v)$. Equation (92) states that

- If v is not an active node ($\eta_v = 0$) and the flow on its active link γ_v is within capacity, then $\frac{d\delta_v^0}{dt} = 0$.

- If v is an active node ($\eta_v = 1$) or it's active link γ_v is over capacity,

$$\left.\frac{d\delta_v^0}{dt}\right|_t = \begin{cases} \dfrac{\sum_{p\in\Gamma_v^0(t)} \lambda_p^0(t)}{Q_v^0(t)} - 1 \\ \qquad\qquad \text{if } v \text{ is the first active queue} \in p \\ \dfrac{\sum_{p\in\Gamma_v^0(t)} \lambda_p^0(t)}{Q_v^0(t)} - \dfrac{\sum_{p\in\hat{\Gamma}_v^0(t)} \lambda_p^0(t)}{\hat{Q}_v^0(t)} \text{ otherwise} \end{cases} \tag{102}$$

As all the variables in equation (102) other than the flow $\lambda_p^0(t)$ are constant during an elementary segment $T^e(v)$ and the flow $\lambda_p^0(t)$ is continuous in t for all $t \in T^e(v)$, we can show that equation (102) admits a unique solution on the interval $T^e(v)$ by the Picard-Lindelöf theorem. □

We have now all the ingredients to prove Theorem 4.

Proof of Theorem 4: We will prove the following proposition: *The time interval of interest $(t_{initial}, t_{final}]$ can be partitioned into a finite set of elementary segments, and the solution to problem (2) exists and is unique*

The proof is done inductively over the depth of the network. If the network contains a single node v_0, $[t_{\text{initial}}, t_{\text{final}}]$ is an elementary segment for v_0, $(t_{\text{initial}}, t_{\text{final}}] \in T^e(v_0)$ and there is a unique solution by Lemma 15. By the induction hypothesis, let us now assume that $(t_{\text{initial}}, t_{\text{final}}]$ can be partitioned into a finite number of elementary segments with respect to all nodes of depth n and that the solution exists and is unique. Let $t_0, t_1, \cdots, t_m$ be times such that $E_n = \{(t_i, t_{i+1}], \forall i \in [0, m-1]\}$ is the set of elementary segments for nodes of depth n, and let δ_v for all $v \in \{V | d(v) \leq n\}$ be the unique solution of depth n.

Let K_n be the non-empty set of nodes of depth n, and let $v \in K_n$ be a node in this set. Lemma 14 gives that for each $v \in K_n$, there is at most one state transition on $(t_i, t_{i+1}]$. Let $F_n(v)$ be the set of times at which these transitions occur for node v. Since there are m elementary segments, there can at most be $|F_n(v)| \leq m$ transitions. If F_n is the set of times at which the transitions for all nodes of depth n happen, $|F_n| \leq m \cdot K_n$.

Let $\{t'_0, t'_1, \cdots, t'_{m'}\} = \{t_0, t_1, \cdots, t_m\} \cup F_n$ be the m' segments created by splitting E_n at each of the state transitions for nodes of depth n. The total number of segments m' satisfies $m' \leq m \cdot (K_n + 1)$, since $|F_n| \leq m \cdot K_n$. By the definition of the t'_i, for each $i \in [0, m']$ we have

- for all $v \in K_n$, η_v is constant on $(t'_i, t'_{i+1}]$,
- $(t'_i, t'_{i+1}]$ is a segment of constant constraint J, since it is subset of an elementary segment, which is already by definition a segment of constant constraint.

Thus, $[t'_i, t'_{i+1}]$ is an elementary segment for all nodes of depth $(n+1)$. Furthermore, by Lemma 15, this implies that there is an unique solution to all nodes of depth $n+1$, which concludes the proof. □

This also completes the proof of Theorem 1.

Proof of Theorem 1: Problem 1 is equivalent to Problem 2 by Theorem 3 and Problem 2 admits a unique solution by Theorem 4. □

In some applications, it is also important to be able to computing the total delay experienced by an agents that takes a particular path. A provides analytical expressions for the total delay along a path.

IV. Applications

The solution to problem (1) models the flows in the network given the departure time functions at the origin and the initial delays by providing the departure time functions for all the other nodes in the network. The solution can be obtained by first solving problem (2), which provides the agent delay function at each node. Practically, problem (2) easier to directly solve than problem (1), because it corresponds to an explicit automaton that is easy to implement for numerical simulations.

Given a discretization time step Δt and the initial conditions $\delta_v(0)$, algorithm 1 gives a numerical solution to the discretized problem (2). The algorithm numerically integrates the ordinary differential equation (ODE) given in equation (92) over time to obtain the solution. The algorithm relies on the fact that each discretized time step is an elementary segment, because the path flows and capacities are assumed to be constant (discrete approximation) during each time step.

Algorithm 1 Calculate approximate solution of problem (2)

solveDelays(*sourceFlow:* λ^0, *initialDelays:* $\delta^0[0]$, *capacities:* μ)

  for $l \in L_0^{\text{out}}$ **do**
    for $t = 1$ to T **do**
      update$(v_l^{\text{out}}, t, 1, 0)$
    end for
  end for

  update(*node:* v, *timeStep:* t, *lastActiveConstraint:* $\hat{\omega}$)
  if $v \notin S$ **then**
    $\Delta_{0,v}^0[t] = \Delta_{0,\pi_v}^0[t] + \delta_v^0[t-1]$
    for $l \in L_v$ **do**
      $\mu_l^0[t] = \mu_l(t + \Delta_{0,v}^0[t])$
      $c_l^0[t] = \dfrac{\sum_{p\in P_l} \lambda_p^0[t]}{\mu_l^0[t]}$
    end for
    $\gamma_v[t] = \arg\max_{l\in L_v^{\text{out}}} c_{v,l}(t)$
    $\Gamma_v[t] = P_{\gamma_v(t)}$
    $\omega_v[t] = \dfrac{\sum_{p\in\Gamma_v[t]} \lambda_p^0[t]}{\mu_l^0[t]}$
    $\delta_v^0[t] = max\,(0,\ (\omega_v - \hat{\omega}) \cdot \Delta t)$
    for $l \in L_v^{out}$ **do**
      if $\delta_v^0[t] > 0$ **then**
        $update(v_l^{\text{out}},\ t,\ \omega_v)$
      else
        $update(v_l^{\text{out}},\ t,\ \hat{\omega})$
      end if
    end for
  end if

A. Single route with multiple bottlenecks

The first case we will study is that of a simple single path network with multiple queues due to several capacity bottlenecks, as illustrated in figure 3. This network can be modeled as a tree with a single sink, i.e. a single path. Thus, we will remove the path index from the notation in this section. Each internal node v has a unique child, thus the internal nodes can be indexed by the integers $v_0, \cdots, _v n$ and the unique path of the tree is $[v_0, _v 1, \cdots, v_n, v_s]$. Moreover, as they model a succession of queues on the same road, we can assume that the capacity of each link (v_i, v_{i+1}) is constant and equal to capacity of the corresponding road segment $\forall v, \mu_{v_i, v_{i+1}} = \mu$. From theorem 5, we know that the evolution of delay is given by

$$\left.\frac{d\Delta_p^0}{dt}\right|_t = \begin{cases} \dfrac{\sum_{p' \in \tilde{\Gamma}_p^0(t)} \lambda_{p'}^0(t)}{\tilde{Q}_p^0(t)} - 1 & \text{if p has an active queue} \\ 0 & \text{otherwise} \end{cases} \tag{103}$$

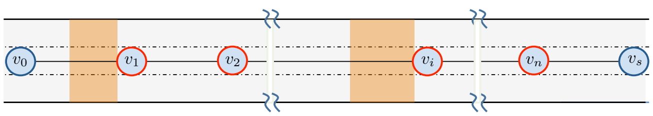

Fig. 3. MultipleBottlenecks on a road.

Since, the link with the smallest capacity will always be the last active link $\tilde{\mu} = \min(\mu_{v_i, v_{i+1}})$:

$$\left.\frac{d\Delta^0}{dt}\right|_t = \begin{cases} \dfrac{\lambda^0(t)}{\tilde{\mu}} - 1 & \text{if there is an active queue} \\ 0 & \text{otherwise} \end{cases} \tag{104}$$

Thus, the evolution of total delay is equivalent to the evolution law for one queue of capacity $\tilde{\mu}$, and the network can be simplified to a unique internal node v followed by a link of capacity $\tilde{\mu}$.

If the capacity of the links is time varying and $\bar{\mu}(t)$ is the capacity of the most constrained link that the agent entering the network at time t is subjected to,

$$\left.\frac{d\Delta^0}{dt}\right|_t = \begin{cases} \dfrac{\lambda^0(t)}{\bar{\mu}(t)} - 1 & \text{if p has an active queue} \\ 0 & \text{otherwise} \end{cases} \tag{105}$$

B. Off-Ramp bottleneck

The next application is to compute the the dynamics of a congested freeway off-ramp, using the off-ramp model presented by Newell [11]. This example shows the versatility of our framework, since Newell's the model includes non-FIFO dynamics at the off-ramp. This is accommodated by introducing an additional node and state dependent capacities on two links. The description of the model is as follows. As seen in figure 4(a), there are two flows λ_h and λ_e that enter the network, which has a capacity of μ_h. Therefore, $\lambda_h(t) + \lambda_e(t) \leq \mu_h$. The exiting flow λ_e is restricted by a capacity constraint of μ_e at the exit. There are four possible states of queuing dynamics that can occur based on the flow values. Figure 5 illustrates the transitions between the states.

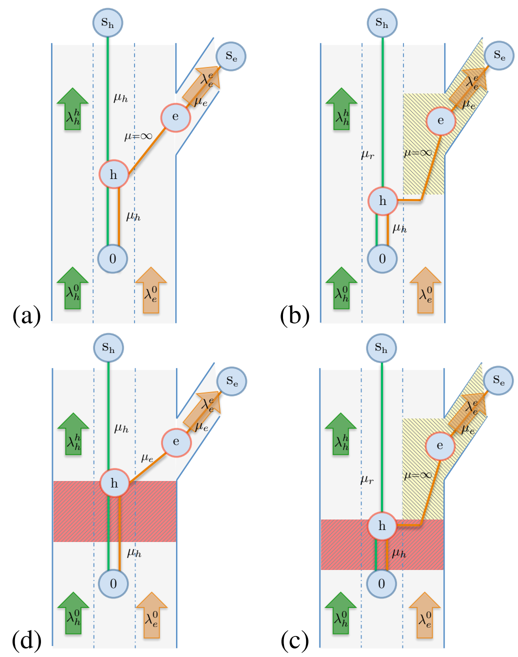

Fig. 4. Off-Ramp model - (a) state 00 (b) state 01 (a) state 10 (a) state 11

Case 1: $\lambda_e \leq \mu_e$. If $\lambda_e(t) \leq \mu_e$, no queues will form in the network and there will be no delay.
Case 2: $\lambda_e > \mu_e$ and $\lambda_h \leq \mu_r$. If $\lambda_e(t) > \mu_e$, an exit queue will start forming at the entrance to the exit as seen in figure 4(b), which will then restrict the capacity of the freeway from μ_h to μ_r.
Case 3: $\lambda_e > \mu_e$, $\lambda_h > \mu_r$ and $\frac{\mu_r}{\lambda_h} \cdot \lambda_e \geq \mu_e$. If the freeway flow $\lambda_h > \mu_r$, then a second freeway queue will start forming behind the exiting agent queue, as seen in figure 4(c), since the freeway demand is greater than the new reduced freeway capacity μ_r. This second freeway queue will contain both freeway and exiting agents and therefore the flow exiting the queue will be subject to the *first-in-first-out* (FIFO) condition. As a result, since the freeway flow λ_h is restricted to a rate of μ_r, the exiting agent flow at the freeway queue will be restricted to $\lambda_e' = \frac{\mu_r}{\lambda_h} \cdot \lambda_e$.
Case 4: $\lambda_e > \mu_e$, $\lambda_h > \mu_r$ and $\frac{\mu_r}{\lambda_h} \cdot \lambda_e < \mu_e$. Now, if $\lambda_e' < \mu_e$, then the off-ramp queue will start decreasing since the flow is less than the capacity and the queue will disappear. Thus, in this case, an off-ramp bottleneck created a second bottleneck that in turn removed the off-ramp bottleneck, which is an unstable equilibrium. Therefore, as explained in [11], there will be a single queue of both freeway and exiting agents that occurs at the off-ramp, as seen in figure 4(d), and the freeway flow through the bottleneck will be $\lambda_h^{out} = \frac{\mu_e}{\lambda_e} \cdot \lambda_h$ according to the FIFO condition.

The uniqueness and existing properties hold even with the state dependent capacities, since the flows are assumed to be piecewise polynomial and therefore lead to a finite number of state transitions. This implies that the link capacities are piecewise constant. Therefore, we can solve for the delays in this network using algorithm 1. Furthermore, this subnetwork can be part of a larger network over which we wish to compute the system delays.

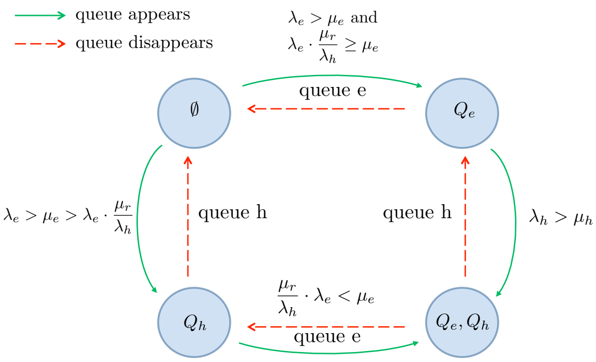

Fig. 5. State transitions in the off-ramp model. The four states $\emptyset$, Q_e, (Q_e, Q_h) and Q_h correspond respectively to the cases (a), (b), (c) and (d) from figure 4.

Figure 6 shows the flow and delay profiles for a numerical example of the off ramp network with the following link capacities: $\mu_E = 5, \mu_H = 30$ and $\mu = 45$. We can observe the following state transitions during the simulated time window.

- At `t=92` Appearance of exiting agent queue.
- At `t=121` Appearance of freeway queue.
- At `t=222` Disappearance of exiting agent queue.
- At `t=372` Disappearance of freeway queue.

One interesting observation is that freeway congestion caused by the exiting agent bottleneck persists well beyond the time at which the exiting agent queue disappears.

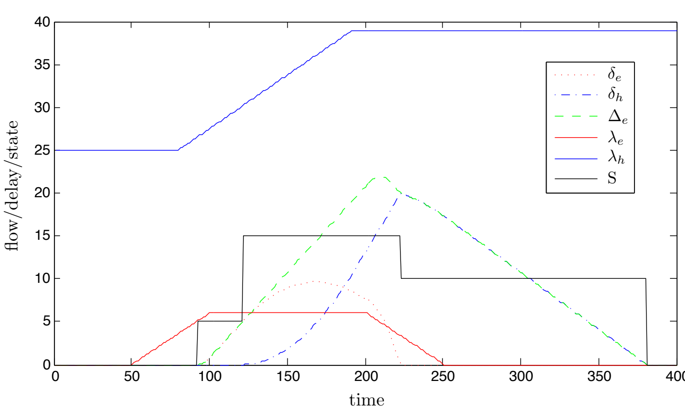

Fig. 6. Simulation of states and delays (δ_E, δ_H) as functions of time t, given the incoming flows at the off ramp, and road parameters: $\mu_E = 5, \mu_H = 30$ and $\mu = 45$

V. Conclusion

This article presented a mathematical framework for modeling traffic flow through a network with a single source and multiple sinks. The model satisfies the standard laws of flow dynamics and is shown to lead to a well-posed dynamics problem with an unique solution. The main benefit of this framework is the ability to analytically prescribe the delays at each junction as a function of the boundary flows at any other upstream junction and the delay over any sub-path with respect to the boundary flow at the source node of the sub-path. This is a critical requirement when solving control and optimization problems over the network, since solving an optimization problem over simulation models is generally intractable in terms of computational complexity. The versatility of computing the delays as a function of the inflow at any point in the network is achieved though a mathematical framework for time mapping the delays. An algorithm for computing a discrete approximation of the system numerically is also provided. The application of the framework is illustrated using two examples, a single path consisting of multiple bottlenecks and a diverge junction with complex junction dynamics. The main limitation of this framework is the limitation to single source networks. The time mapping framework presented can however be generalized to any non-cyclic (tree) network. Thus, the next step would be to introduce merging dynamics into the model to obtain a more general network model.

References

[1] Abdelnaser Adas. Traffic models in broadband networks. *Communications Magazine, IEEE*, 35(7):82–89, 1997.
[2] Cheng-Shang Chang. *Performance guarantees in communication networks*. Springer, 2000.
[3] Carlos F Daganzo. The Cell Transmission Model , Part II Network Traffic. 298(2):79–93, 1995.
[4] Victor S Frost and Benjamin Melamed. Traffic modeling for telecommunications networks. *Communications Magazine, IEEE*, 32(3):70–81, 1994.
[5] Sushant Jain, Kevin Fall, and Rabin Patra. Routing in a delay tolerant network. In *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '04, pages 145–158, New York, NY, USA, 2004. ACM.
[6] Jean-Patrick Lebacque. The godunov scheme and what it means for first order traffic flow models. In *Internaional symposium on transportation and traffic theory*, pages 647–677, 1996.
[7] Hong K. Lo, Elbert Chang, and Yiu Cho Chan. Dynamic network traffic control. *Transportation Research Part A: Policy and Practice*, 35(8):721 – 744, 2001.
[8] Deepak K Merchant and George L Nemhauser. A Model and an Algorithm for the Dynamic Traffic Assignment Problems. *Transportation science*, 12(3):183–199, 1978.
[9] Ian M Mitchell, Alexandre M. Bayen, and Claire J Tomlin. A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games. *Automatic Control, IEEE Transactions on*, 50(7):947–957, 2005.
[10] Anna Nagurney, June Dong, and Ding Zhang. A supply chain network equilibrium model. *Transportation Research Part E: Logistics and Transportation Review*, 38(5):281–303, 2002.
[11] Gordon F. Newell. Delays caused by a queue at a freeway exit ramp. *Transportation Research Part B: Methodological*, 33(5):337 – 350, 1999.
[12] Srinivas Peeta and Athanasios K. Ziliaskopoulos. Foundations of Dynamic Traffic Assignment : The Past , the Present and the Future. *Networks and Spatial Economics*, pages 233–265, 2001.
[13] Jack Reilly, Samitha Samaranayake, Maria Laura Delle Monache, Walid Krichene, Paola Gaotin, and Alexandre M. Bayen. An efficient method for coordinated ramp metering using the discrete adjoint method. *Journal of Optimization Theory and Applications, in review*, 2013.
[14] Samitha Samaranayake, Axel Parmentier, Yiguang Xuan, and Alexandre M. Bayen. Congestion reduction at an off-ramp via incentives for demand shift. *In preparation*, 2013.
[15] Samitha Samaranayake, Jack Reilly, Walid Krichene Maria Laura Delle Monache, Paola Gaotin, and Alexandre M. Bayen. System Optimal Dynamic Traffic Assignment with Partial Compliance (SO-DTA-PC). *Technical report*, 2014. http://dx.doi.org/10.7922/G23X84KV.
[16] Ioannis Simaiakis and Hamsa Balakrishnan. Queuing models of airport departure processes for emissions reduction. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2009.
[17] Harold S Stone. Multiprocessor scheduling with the aid of network flow algorithms. *Software Engineering, IEEE Transactions on*, (1):85–93, 1977.
[18] Marshall H Stone. The generalized weierstrass approximation theorem. *Mathematics Magazine*, 21(5):237–254, 1948.
[19] Dan Work, Sebastien Blandin, Olli-Pekka Tossavainen, Benedetto Piccoli, and Alexandre M. Bayen. A traffic model for velocity data

assimilation. *AMRX Applied Mathematics Research eXpress*, 1:1–35, 2010.

Appendix A
Total path delay

In some applications, it is also important to be able to computing the total delay experienced by an agents that takes a particular path. In this section, we provide analytical expressions for the total delay along a path.

Definition 28. *Total delay of a path p*
We define the total delay Δ_p^0 encountered on a path p at time t as the total delay encountered by agent on path p that enters on the network at t throughout its entire path to the sink node.

$$\Delta_p^0(t) = [D_p^{v_{p_N}}]^{-1}\left(D_p^0(t)\right) - t \tag{106}$$

where v_{p_N} is the last non-sink node on path p. We define the time mapped total delay Δ_p^j as the total delay in path p as seen by an agent that is at node j at time t.

$$\Delta_p^j = \mathbf{T}^{j,0}\left(\Delta_p^0\right) \tag{107}$$

Proposition 16. *Total delay Δ_p^j as a function of queue delay δ*
The time mapped total delay Δ_p^j encountered on a path is equal to the sum of delay encountered by the agent on its path.

$$\Delta_p^j\left(t^j\right) = \sum_{v \in V_p \setminus (\{0\} \cup S)} \delta_v^j\left(t^j\right) \tag{108}$$

where t^j is the time that the agent is at node j.

Proof: Let $t^i = T^{i,j}\left(t^j\right)$. We obtain the result as follows using the definition of delay and a series of time mappings.

$$\begin{aligned} LHS &= \Delta_p^j\left(t^j\right) \\ &= \mathbf{T}^{j,0}\left(\Delta_p^0\left(t^j\right)\right) \\ &= \Delta_p^0\left(T^{0,j}\left(t^j\right)\right) \\ &= \Delta_p^0\left(t^0\right) \\ &= [D_p^{v_{p_N}}]^{-1}\left(D_p^0\left(t^0\right)\right) - t^0 \end{aligned}$$

$$\begin{aligned} RHS &= \sum_{v \in V_p \setminus (\{0\} \cup S)} \delta_v^j\left(t^j\right) = \sum_{v \in V_p \setminus (\{0\} \cup S)} \delta_v^{\pi_v}\left(t^{\pi_v}\right) \\ &= \sum_{v \in V_p \setminus (\{0\} \cup S)} [D_p^v]^{-1}\left(D_p^{\pi_v}\left(t^{\pi_v}\right)\right) - t^{\pi_v} \\ &= \sum_{v \in V_p \setminus (\{0\} \cup S)} [D_p^v]^{-1}\left(D_p^{\pi_v}\left(t^{\pi_v}\right)\right) - [D_p^{\pi_v}]^{-1}\left(D_p^{\pi_v}\left(t^{\pi_v}\right)\right) \\ &= \sum_{v \in V_p \setminus (\{0\} \cup S)} [D_p^v]^{-1}\left(D_p^0\left(t^0\right)\right) - [D_p^{\pi_v}]^{-1}\left(D_p^0\left(t^0\right)\right) \\ &= [D_p^{v_{p_N}}]^{-1}\left(D_p^0\left(t^0\right)\right) - [D_p^0]^{-1}\left(D_p^0\left(t^0\right)\right) \\ &= [D_p^{v_{p_N}}]^{-1}\left(D_p^0\left(t^0\right)\right) - t^0 \end{aligned}$$

■

Definition 29. *Active link of the last active queue of a path p at time t ($a_p(t)$)*
Let p be a path and t be the time that an agent departs node j. We define the last active queue of the path p time mapped to passing node j at time t as

$$a_p^j(t) = \max_{\succ}\left\{v \in V_p | \eta_v^j(t) = 1\right\} \tag{109}$$

For notational convenience we also define the following:

$$\begin{aligned} \tilde{\gamma}_p^j(t) &= \gamma^j_{a_p^j(t)}(t) && (110) \\ \tilde{\Gamma}_p^j(t) &= \Gamma^j_{a_p^j(t)}(t) && (111) \\ \tilde{Q}_p^j(t) &= \mu^j_{\tilde{\gamma}_p^j(t)}(t) && (112) \end{aligned}$$

Theorem 5. ***Evolution law for total delay*** Δ_p^0
Let p be a path, t be a time. The evolution law for total delay at time t is

$$\left.\frac{d\Delta_p^0}{dt}\right|_t = \begin{cases} \dfrac{\sum_{p' \in \tilde{\Gamma}_p^0(t)} \lambda_{p'}^0(t)}{\tilde{Q}_p^0(t)} - 1 & \text{if } p \text{ has an active queue} \\ 0 & \text{otherwise} \end{cases} \tag{113}$$

Proof: Taking the derivative of equation (108) for $j = 0$, we obtain

$$\left.\frac{d\Delta_p^0}{dt}\right|_t = \sum_{v \in V_p \setminus (S \cup \{0\})} \left.\frac{d\delta_v^0}{dt}\right|_t \tag{114}$$

$$= \sum_{\{v | v \in V_p \setminus (S \cup \{0\}), \gamma_v^0(t) = 1\}} \left.\frac{d\delta_v^0}{dt}\right|_t \tag{115}$$

Note that $\gamma_v^0(t) = 1$ implies node v is active when the source flow at time t reaches node v. From theorem 2 with $j = 0$ we have,

$$\left.\frac{d\delta_v^0}{dt}\right|_t = \begin{cases} \dfrac{\sum_{p \in \Gamma_v^0(t)} \lambda_p^0(t)}{Q_v^0(t)} - 1 & \text{if } v \text{ is the first active queue} \in p \\ \dfrac{\sum_{p \in \Gamma_v^0(t)} \lambda_p^0(t)}{Q_v^0(t)} - \dfrac{\sum_{p \in \hat{\Gamma}_v^0(t)} \lambda_p^0(t)}{\hat{Q}_v^0(t)} & \text{otherwise} \end{cases} \tag{116}$$

Plugging this into equation (115) gives a telescopic series, since it only considers the active nodes of the path and $\tilde{Q}_p^0(t)$ gives the capacity of the last active link of path p. Thus, we obtain

$$\left.\frac{d\Delta_p^0}{dt}\right|_t = \frac{\sum_{p' \in \tilde{\Gamma}_p^0(t)} \lambda_{p'}^0(t)}{\tilde{Q}_p^0(t)} - 1 \tag{117}$$

If p does not contain an active queue there is no queuing in the path, which means there is no change in the queue length and therefore no change in the delay. □

Remark 9. *Note that this theorem can be extended to any subpath $p_{ij} \in p$ such that*

$$\left.\frac{d\Delta_{p_{i,j}}^i}{dt}\right|_t = \frac{\sum_{p' \in \tilde{\Gamma}_{p_{ij}}^i(t)} \lambda_{p'}^i(t)}{\tilde{Q}_{p_{ij}}^i(t)} - 1 \tag{118}$$