



# Predicting Traffic Load in Public Transportation Networks

Stefan Haar, Simon Theissing

► **To cite this version:**

Stefan Haar, Simon Theissing. Predicting Traffic Load in Public Transportation Networks. 2016  
American Control Conference, Jul 2016, Boston, United States. <hal-01329632>

**HAL Id: hal-01329632**

**<https://hal.inria.fr/hal-01329632>**

Submitted on 9 Jun 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Predicting Traffic Load in Public Transportation Networks

Stefan Haar and Simon Theissing

MExiCo team, INRIA and LSV, CNRS & ENS de Cachan,  
Cachan, France

**Abstract.** This work is part of an ongoing effort to understand the dynamics of passenger loads in modern, multimodal transportation networks (TNs) and to mitigate the impact of perturbations, under the restrictions that the precise number of passengers in some point of the TN that intend to reach a certain destination (i.e. their distribution over different trip profiles) is unknown. We introduce an approach based on a stochastic hybrid automaton model for a TN that allows to compute how such probabilistic load vectors are propagated through the TN, and develop a computation strategy for forecasting the network's load a certain time in the future.

**Keywords:** Stochastic Hybrid Automata, Transportation Networks

## 1 INTRODUCTION

We continue here the work begun in [1] for capturing both the discrete vehicle movements and continuous passenger transfers in a multimodal public transportation network (TN). In [1], a deterministic hybrid automaton (DHA) model was used, in order to overcome the state space explosion problem via fluidification. For its specification we were using discrete and continuous Petri nets (PNs) as the basic modelling blocks [2], with the feature that the marking of the continuous places and thus the flows in-between were not restricted to scalars. Instead, we were integrating different passenger profiles into vector markings and -flows, together with routing matrices relating both.

Now - and this is the starting point of the present work - a *real* TN is everything but deterministic. On the one hand, there are highly unpredictable asynchronous events for which statistical data is hard to obtain. It is thus difficult to include them in the daily network operation [3], e.g. by means of minute-by-minute or hourly forecasts. Typical examples are passenger incidents. Now, note that - apart from few exceptions - such incidents originate locally, in one mode or line, and then propagate to other modes or lines by passenger transfers. And these transfers are predictable, not necessarily deterministic, if one knows the destination or trip profile of the passengers; in general, this can only be known through probabilistic estimates. Finally, there are the more "continuous" passenger arrival processes for which statistical data is easier to obtain:

How many passenger will arrive at a station at which time? According to which route, including which vehicle missions, will they travel?

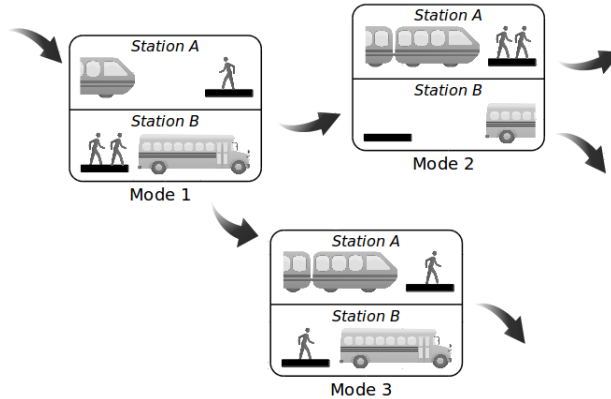


Fig. 1: A discrete mode of the SHA model is defined by the (discrete) state of the vehicle operation. Each mode enables a set of possible passenger transfers and thus the framework for the continuous dynamics associated with it. State-dependent conditions trigger forced jumps between the modes.

Here, we will extend the DHA from [1] in that we will replace all deterministic passenger arrival processes by their stochastic counterparts, and, in doing so, introduce a Markovian stochastic hybrid automaton (SHA) with jumps between the discrete modes at a priori defined equidistantly spaced discrete points in time. We will then informally define a computation strategy for the forecast of a TN's load in form of a probabilistic forward reachability problem. A look into the literature then reveals that our SHA model knows many preceding - similar or alternative - modelling approaches that have been developed notably in the past two decades, with every approach introducing the uncertainty at a different point in the model dynamics. For instance, the authors of [4] extended the dynamics underlying a DHA in that the jumps between the discrete modes are either exponentially distributed functions of time or immediate; with a weighting function as a means to resolve conflicts among simultaneously enabled immediate transitions [5]. Thus, as compared to our modelling approach, the jumps in [5] are not confined to a countable set of discrete points in time. However, in their modelling approach the discrete jumps are decoupled from the continuous states, and the latter evolve according to deterministic differential balance equations; whereas in our model the continuous states evolve according to stochastic differential equations (SDEs) and jumps might be coupled to the continuous states. The authors of [6] bridge the gap between the continuous states and the discrete jumps by means of a guard function. Finally, the deterministic balance equations were replaced by

normally distributed balance equations in [7]. Outside the framework of PNs, the authors of [8] introduced an SHA that exhibits state-driven (forced) jumps between the discrete modes subjected to sets of SDEs; for each mode one set. This approach was extended in [9] in that the mode transitions are no longer restricted to forced jumps, but can be initiated by spontaneous jumps with state-dependent transition rates as well. The author of [10], then showed how the SHA from [9] can be formulated in an equivalent system of integro-differential equations together with boundary conditions. Regarding the probabilistic forward reachability problem, note that the authors of [11] presented a grid-based asymptotic approximation method for a backward reachability problem subjected to the dynamics of an SHA that encounters spontaneous jumps between the discrete modes; with a system of SDEs assigned to every mode. That system is approximated by a Markov chain following a space and subsequent time discretization; whereas in our approach the discretization of the time precedes the discretization of the space, and the latter comes along with a numerical integration of the continuous states in a discrete mode.

In the rest of this paper, we will introduce the basic ingredients for our SHA model in Sec. 2, on top of which we will elaborate in Sec. 3 the continuous dynamics for a particular discrete mode in it (cf. Fig. 1). We will then define a network’s vehicle load in Sec. 4 that we will use in Sec. 5 for computing forecasts.

## 2 BASIC INGREDIENTS

### 2.1 Infrastructure

The operation of all vehicles ( $\mathcal{V}$ ) involving a finite set of runs ( $\mathcal{Z}$ ) and missions ( $\mathcal{X}$ ), and the routing of all passengers according to a finite set of trip profiles ( $\mathcal{Y}$ ), define the dynamics of our SHA model. They are specified on a shared infrastructure comprising a finite set of stations ( $\mathcal{S}$ ), another finite set of transportation grids ( $\mathcal{G}$ ), and yet another finite set of interfaces ( $\mathcal{I}$ ) connecting grids and stations. We will informally introduce first this infrastructure, and then the vehicle operation and the passenger routing. A more detailed description can be found in [1].

Every *transportation grid* (TG)  $g \in \mathcal{G}$  confines the lines and thus the vehicle movements of one particular mode. It comprises a finite set of *waypoints* and routes connecting them, in which a particular waypoint refers to a distinct discrete vehicle position and a route between two waypoints to the possibility of a vehicle transfer between both. A waypoint is either empty or accommodates one vehicle at a time. A rule set then defines a deterministic resolution between conflicting vehicle movements.

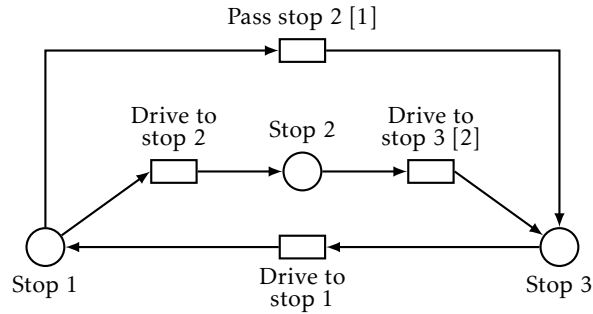


Fig. 2: Sample transportation grid

Fig. 2 shows a simple TG that may accommodate two orbital metro (sub-)lines: A vehicle at the waypoint “Stop 1” can either drive to the waypoint “Stop 2” and then to the waypoint “Stop 3”, or it can pass the second stop. Regarding the two integers “1” and “2” written in brackets next to “Pass stop 2” and “Drive to stop 3”, they indicate that a vehicle movement from the first stop to the third stop is privileged over a conflicting vehicle movement from the second stop to the third stop.

Every *station*  $s \in \mathcal{S}$  can be decomposed into a finite set of *gathering points* (GPs) that are connected by the possibility of passenger transfers to other GPs, the outdoor area of the TN, or the waypoints in the TGs. They accommodate the transferring passengers and have limited capacities.

As an example, Fig. 3 depicts an extract of a sample station: Before joining the station’s entrance area (= GP), the passengers have to step down the stairs (= possibility of a passenger transfer) starting from the TN’s outdoor area. They can then cross the turnstiles so as to transfer to the platform that itself is dedicated to the boarding of a vehicle executing mission  $x \in \mathcal{X}$ . Moreover, the entrance area can accommodate 50 passengers, and the platform 200.

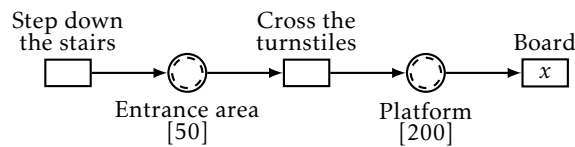


Fig. 3: Extract of a sample station

Finally, every *interface*  $i \in \mathcal{I}$  establishes the link between a GP of a station and a waypoint in a TG, or vice versa (cf. 4): A passenger on-board a vehicle stopped at the waypoint “Stop 1” and executing the mission  $x_1 \in \mathcal{X}$  can alight from it to the GP “Platform 1” from which s/he can either exit the station or

transfer to the GP “Platform 2”. Then, from the “Platform 2” a passenger can board a vehicle iff that vehicle is executing the mission  $x_2 \in \mathcal{X}$  and stopped at the waypoint “Stop 2”.

## 2.2 Vehicle operation

Every *vehicle mission*  $x \in \mathcal{X}$  defines a path in a TG, a sequence of stops at the waypoints, and deterministic driving times in between: The vehicle stops are specified by pairs of minimum and maximum dwell times that are assigned to every waypoint. In particular, a positive minimum dwell time indicates that a vehicle has to stop at a waypoint. A positive maximum dwell time, on the contrary, applies iff a vehicle’s intended movement is not conflicting with that of another one, i.e., a vehicle movement with a higher priority. Now every *vehicle run*  $z \in \mathcal{Z}$  specifies a sequence of vehicle missions, and assumes a vehicle realizing it to be in the operational state “Parking” shortly before and afterwards. Finally, the *dispatch plan*  $\mathcal{P}_d$  defines which vehicle is supposed to start realizing which vehicle run at which time. The emphasis here is on “supposed” since the dispatch of a vehicle at a particular time obviously requires that the vehicle is available, i.e., parked at the correct waypoint.

## 2.3 Passenger routing

In the infrastructure from Fig. 4, we decorate every arc with a subset of the set  $\mathcal{Y} = \{1, 2, 3, 4\}$  of trip profiles (with set brackets omitted) for their specification.

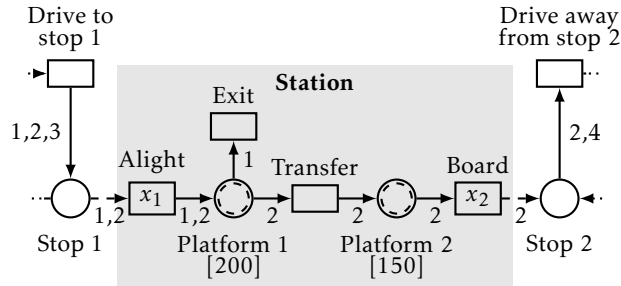


Fig. 4: Passenger transfer in a station between two lines including the graphical specification of four trip profiles

Every *trip profile*  $y \in \mathcal{Y}$  fixes routing of the corresponding group of passengers in the TN along a tree in the infrastructure. That way one can account for alternative vehicle missions in the routing of the same group of passengers. For instance, a particular trip profile might be specified s.t. a passenger at the platform of a station will only board one particular vehicle mission, whereas a passenger, who is travelling according to another trip profile, is taking whatever

vehicle mission is arriving next. Once having specified all trip profiles, they can then be mapped to square matrices assigned to all possible passenger transfers in the infrastructure. We can also integrate the possibility of re-routing into these routing matrices; such a re-routing means that passenger change their trip profiles [1].

### 3 PASSENGER FLOWS IN A MACRO STATE

#### 3.1 Macro states

At any given time, the TN is in one of its discrete states in terms of the vehicle operation (where are the vehicles, are they stopped in a station allowing boarding, or moving between stations etc); we call these untimed discrete states *macro states* of the SHA. That is, a macro state gives (i) the positions of all vehicles in form of discrete waypoints in the TGs, (ii) which vehicles are in operation, (iii) which of them are stopped, and (iv) the run, together with the current mission, that each vehicle is executing.

**Definition 1.** A macro state of a TN =  $\{\mathcal{G}, \mathcal{S}, \mathcal{I}, \mathcal{V}, \mathcal{X}, \mathcal{Z}, \mathcal{Y}, \mathcal{P}_d\}$  is a tuple  $M := (\mathcal{V}_+, \mathcal{V}_-, \nu_p, \nu_r, \nu_x)$ , where

- $\mathcal{V}_+ \subseteq \mathcal{V}$  is the set of vehicles in operation,
- $\mathcal{V}_- \subseteq \mathcal{V}_+$  is the set of stopped vehicles,
- $\nu_p: \mathcal{V} \rightarrow \bigcup_{g \in \mathcal{G}} P^{(g)}$  gives the position  $\nu_p(v)$  of  $v \in \mathcal{V}$ , in which  $P^{(g)}$  denotes the set of all waypoints in  $g \in \mathcal{G}$ ,
- $\nu_r: \mathcal{V}_+ \rightarrow \mathcal{Z}$  gives the run  $\nu_r(v)$  of  $v \in \mathcal{V}_+$ , and
- $\nu_x: \mathcal{V}_+ \rightarrow \mathcal{X}$  gives the mission  $\nu_x(v)$  of  $v \in \mathcal{V}_+$ .

Thus, every macro state uniquely fixes all passenger flows that are possible in it: the flow of passengers entering the system, and the flows between the GP's in the stations and the stopped vehicles, in which we assume that the passengers on-board a vehicle can alight from it iff the vehicle is stopped at a waypoint, and, according to the specification of all interfaces, that waypoint is connected to a GP in a station. Similarly, we also assume that a passenger at a platform can board a vehicle iff that vehicle is stopped at a waypoint that can be accessed from the platform.

#### 3.2 Balance equations

The continuous dynamics in any macro state is given by a system of decoupled stochastic differential (Itô) equations for time  $\tau \geq 0$ , of the form

$$dX_s(\tau) = A_s(X_s(\tau)) d\tau + B_s dW(\tau), \quad (1)$$

with one such equation for every station  $s \in \mathcal{S}$ . In Eqn. 1,  $X_s$  is the state vector of the passenger loads of (i) every GP in  $s$ , and (ii) every vehicle *docked* to  $s$ ,

i.e., every  $v \in \mathcal{V}_-$  that can be accessed from a GP in  $s$  due to the specification of a particular interface. Assuming that there are  $c_1 \in \mathbb{N}_{>0}$  GPs in  $s$  and  $c_2 \in \mathbb{N}_{\geq 0}$  vehicles docked to  $s$ , then  $X_s \in \mathbb{R}^m$ , in which  $m := (c_1 + c_2)n$ , and  $n := |\mathcal{V}|$  denotes the number of different trip profiles.  $A_s$  is the drift of  $X_s$ , and  $B_s$  its diffusion. Finally,  $W$  denotes an  $n$ -dimensional Wiener process<sup>1</sup> [12]. Again, note that the equations for different stations are decoupled.

We obtain Eqn. 1 by setting up the balance equations for all passenger loads contributing to  $X_s$ . In this context, we note that we do not have to set up separate balance equations for the complementary free capacities since they can be deduced from the capacity limits of the GPs and the vehicles, respectively. Thus, if (i)  $P_s$  denotes the set of all GPs in  $s$ , (ii)  $\mathcal{V}_s$  the set of all stopped vehicles that are docked to  $s$ , and (iii)  $m_s(k, \tau)$ , with  $m_s : P_s \cup \mathcal{V}_s \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$ , the  $n$ -dimensional passenger load of a GP  $k \in P_s$  or a vehicle  $k \in \mathcal{V}_s$ , we can then compute its complementary free capacity  $m'_s(k, \tau)$ , with  $m'_s(k, \tau) : P_s \cup \mathcal{V}_s \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$ , from the invariant

$$m'_s(k', \tau) = m'_s(k, 0) + 1^T (m_s(k, 0) - m_s(k, \tau)). \quad (2)$$

Therein,  $1$  denotes an  $n$ -dimensional column matrix with ones only, and  $1^T$  its transpose. Moreover, note that the domains for passenger loads and free capacities are not restricted to the positive reals in Eqn. 1; a constraint that we will explicitly enforce in its numerical computation. That being said, the balance equation for the passenger load of every  $k \in P_s \cup \mathcal{V}_s$  at  $\tau \geq 0$  reads

$$\begin{aligned} dm_s(k, \tau) := & \sum_{t \in \bullet k} R_s(t) (\phi_s(t, \tau) d\tau + D_s(t) dW(\tau)) \\ & - \sum_{t \in k \bullet} \phi_s(t, \tau) d\tau. \end{aligned} \quad (3)$$

Equation 3 relates the temporal change of  $m_s(k, \tau)$  to the diffusion, passenger flow, and routing matrix that come along with every passenger transfer  $t \in \bullet k \cup k \bullet$  joining or leaving it, in which  $\bullet k/k \bullet$  denotes the set of all incoming/outgoing passenger transfers w.r.t.  $k$ . In particular,  $D_s(t)$ ,

$$D_s : T_s \rightarrow \{K \in \mathbb{R}^{n \times n} : K[i, j] = 0 \text{ for any } i \neq j\}$$

specifies the diffusion,  $\phi_s(t, \tau)$ ,  $\phi_s : T_s \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}^n$ , the passenger flow, and  $R_s(t)$ ,

$$R_s : T_s \rightarrow \{K \in [0, 1]^{n \times n} : K[\cdot, i] \in \{0, 1\} \text{ for any } i \in \{1, 2, \dots, n\}\}$$

the routing of  $\phi_s(t, \tau)$  assigned to  $t$  at  $\tau$ ; in which  $K[\cdot, j]$  denotes the  $j$ -th column of a matrix  $K$ , and  $K[i, j]$  the element in the  $i$ -th row thereof.

Now regarding the specification of  $D_s$ , recall that uncertainty enters the TN in form of the passengers arriving at the stations from its outdoor area.

<sup>1</sup> A continuous-time stochastic process with independent and stationary increments  $W_t - W_s$  whose law is Gaussian with parameter  $t - s$



In other words, we do not know how many passengers arrive at which time and according to which trip profile. However, once having entered, the routing of every passenger including the vehicle operation is deterministic. We then demand  $D_s$  to be zero for every passenger transfer (not arrival), i.e. either (i) a transfer in a station between two GPs, or (ii) a transfer between a vehicle docked to a station and a GP, or (iii) a flow of passengers leaving a station from a GP to the outdoor area of the TN. For the arrival processes,  $D_s$  is a tuning parameter that modulates the variances of the multidimensional Wiener process; again, one dimension per trip profile. Then, turning towards  $\phi_s$ , we note that its specification depends on the particular use case as discussed in [1]. Finally,  $R_s$  is deduced from the graphical specification of all trip profiles and the re-routing of the passengers; again, compare the discussion in [1].

### 3.3 Numerical integration

In practice, there are two dominant approaches to computing the temporal change of an initial distribution subjected to Eqn. 1. On the one hand, there is the Monte Carlo method [13]: From the pool of all possible solution paths, some paths are selected “randomly” and possibly several times [14, pp. 274-277]. In this approach, one thus would have to ensure that the correlation between the selected solution paths matches the correlation found in the actual stochastic process. Of course, this is a difficult task especially if the statistical data is not at hand, as is often the case in degraded modes of operation of TNs. The second approach is the one we will use here: Integrate the Itô process from Eqn. 1 into systems of linear parabolic partial differential equations (one for every station  $s \in \mathcal{S}$ ):

$$\begin{aligned} \frac{\partial}{\partial \tau} p_s(x, \tau) = & - \sum_{i=1}^N \frac{\partial}{\partial x_i} (A_s(x)[i] p_s(x, \tau)) + \\ & \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \Psi_s[i, j] \frac{\partial^2}{\partial x_i \partial x_j} p_s(x, \tau), \end{aligned} \quad (4)$$

with  $A_s, B_s$  from (1) and the abbreviation  $\Psi_s := B_s B_s^T$ . This system is also known as the (multidimensional) Fokker-Planck (FP) or the Kolmogorov forward equation, and describes the time evolution of the probability density function (PDF)

$$p_s : \mathbb{R}^m \times \mathbb{R}_{\geq 0} \rightarrow [0, \infty).$$

We then note that by introducing the probability flux

$$f_s(x, \tau) := A_s(x) p_s(x, \tau) - \frac{1}{2} \Psi_s \begin{bmatrix} \frac{\partial}{\partial x_1} p_s(x, \tau) \\ \frac{\partial}{\partial x_2} p_s(x, \tau) \\ \vdots \\ \frac{\partial}{\partial x_m} p_s(x, \tau) \end{bmatrix}, \quad (5)$$

we can rewrite Eqn. 4 and obtain the continuity equation

$$\frac{\partial}{\partial \tau} p_s(x, \tau) + \operatorname{div}(f_s(x, \tau)) = 0 \quad (6)$$

in its differential form. Therein,

$$\operatorname{div}(f_s(x, \tau)) := \sum_{i=1}^N \frac{\partial}{\partial x_i} f_s(x, \tau) \quad (7)$$

denotes the divergence operation applied to the probability flux from Eqn. 5. Now from this continuity equation we can derive reflecting boundary conditions for the numerical integration of Eqn. 4 that confine the probability flux to the closed convex polytope  $K_s \subset \mathbb{R}^m$ . Note that  $K_s$  comprises all those states  $x \in \mathbb{R}^m$  that do not violate the capacity limits of the GPs and vehicles on the one hand, nor the non-negativity of the passenger loads themselves on the other hand. Here, we summarize only the major steps in their derivation in an informal manner, and refer to the literature [15, Chapter 5] for more details: Starting point is the insight that the cumulative probability of  $X_s$  adopting a value in  $K_s$  at any time instant  $\tau \in \mathbb{R}_{\geq 0}$  must be one. Thus, the time derivative of this cumulative probability must be zero. Then, after some transformations employing the divergence theorem, we obtain the reflecting boundary condition

$$f_s(x, \tau) n_s(x) = 0, \forall \tau \in \mathbb{R}_{\geq 0}. \quad (8)$$

Therein,  $n_s(x)$ , with  $n_s : \partial K_s \rightarrow \mathbb{R}^m$ , denotes the orthonormal vector that (i) has its origin at the state  $x \in \partial K_s$ , i.e., the boundary of  $K_s$ , and (ii) is pointing in the outward direction of  $K_s$ .

From this, it will be possible to compute the time evolution of the PDF subjected to Eqn. 1 starting from  $\tau = 0$ , by including the *initial value problem* (IVP) in form of Eqn. 4, Eqn. 5, and Eqn. 8 in a numerical integration scheme such as the Crank-Nicolson method in combination with the multigrid method [16].

At this point, note that the numerical computation of the IVP above introduces a discretization of the passenger load space, which we tried to circumvent by the fluidification of all passenger flows before. Given the high-dimensionality of Eqn. (1), this new discretization might be prohibitive assuming that no further model simplifications are made; on which we are currently working.

## 4 VEHICLE LOAD

So far, we have introduced the basic ingredients for our SHA model, provided a definition for a macro state, and shown how the SDE system from Eqn. 1 encapsulated in every macro state can be integrated into a numerical integration scheme so as to compute the time evolution of an initial PDF regarding the passenger loads at every GP in a station and on-board every vehicle docked to that

station. Now we are going to integrate the numerical integration in a macro state into a computation strategy, which takes an initial estimation of the TN's load and propagates it into a target set, for prediction of the load's distribution in some time horizon.

The network load can be decomposed into a vehicle load and a passenger load, for every GP in every station and every vehicle; we introduce the vehicle load as a separate mathematical object.

**Definition 2.** A vehicle load of a TN =  $\{\mathcal{G}, \mathcal{S}, \mathcal{I}, \mathcal{Z}, \mathcal{X}, \mathcal{Y}\}$  is a tuple  $L := (M, v_{\text{dr}}, v_{\text{dw}})$ , where

- $M$  is a macro state according to Def. 1,
- $v_{\text{dr}} : \mathcal{V}_+ \setminus \mathcal{V}_- \rightarrow \mathbb{R}_{\geq 0}$  gives the elapsed driving time  $v_{\text{dr}}(v)$  of  $v \in \mathcal{V}_+ \setminus \mathcal{V}_-$  on its way to  $v_p(v)$ , and
- $v_{\text{dw}} : \mathcal{V}_- \rightarrow \mathbb{R}_{\geq 0}$  gives the elapsed dwell time  $v_{\text{dw}}(v)$  of  $v \in \mathcal{V}_-$  at  $v_p(v)$ .

Thus, a vehicle  $v \in \mathcal{V}_+ \setminus \mathcal{V}_-$  with its discrete position pointing to a waypoint  $p$  in a transportation grid, is not physically located at  $p$ . Rather, it is on its way to  $p$  (from its previous discrete position) but has not yet arrived.

**Definition 3.** A network load of a TN =  $\{\mathcal{G}, \mathcal{S}, \mathcal{I}, \mathcal{Z}, \mathcal{X}, \mathcal{Y}\}$  is a tuple  $Q := (L, m)$ , where

- $L$  is a vehicle load according to Def. 2, and
- $m : \bigcup_{s \in \mathcal{S}} P_s \cup \mathcal{V} \rightarrow \mathbb{R}_{\geq 0}^n$  gives the passenger load  $m(a)$  of  $a \in P_s$  in  $s \in \mathcal{S}$  or  $a \in \mathcal{V}$ , in which  $P_s$  denotes the set of all GPs in  $s$ .

The prediction of the TN's load at a time instant  $\tau_0 \in \mathbb{R}_{\geq 0}$  requires the computation of the probability that this network load is contained in some closed region, which is a function of time to be treated with caution. For instance, we might be interested in the probability according to which a platform will be overcrowded within half an hour and half an hour plus  $r \in \mathbb{N}_{>0}$  minutes, starting from  $\tau_0$ . Assuming this to be true, we are then not interested in the network load at any time instant  $\tau \in [\tau_0, \tau_0 + 30 \text{ min}] \cup (\tau_0 + 30 \text{ min} + r \text{ min}, \infty)$ . In this context, note that we can always deduce the prediction horizon  $h := \max_{\Delta} \mathcal{H}$  from the set  ${}_{\Delta} \mathcal{H} \subset 2^{\mathbb{R}_{\geq 0}}$ , in which  $2^{\mathbb{R}_{\geq 0}}$  denotes the power set of  $\mathbb{R}_{\geq 0}$ , of all -overlapping or non-overlapping - time intervals confining the network load to a closed region as a function of time that we will call the *target region*; providing us an upper bound for our computation strategy: Starting from some initial network load at  $\tau_0$ , we will allow the TN to change its vehicle load (= jump of the SHA's discrete state) only at any discrete point in time  $\tau \in \mathcal{H}$ , in which  $\mathcal{H} := \{\tau_0 + \Delta\tau, \tau_0 + 2\Delta\tau, \dots, \tau_0 + \gamma\Delta\tau\}$ ,  $\gamma := \max\{i \in \mathbb{N} : i\Delta\tau \leq h\}$ , and  $\Delta\tau > 0$  denotes the resolution of the prediction. However, we might not start from one particular network load, but from a set of initial network loads - denoted by  $\mathcal{L}_0$  - instead; in that the possibility of being present in any  $L \in \mathcal{L}_0$  at  $\tau_0$  is positive. Accordingly, the possibility of being present in any other vehicle load  $L' \notin \mathcal{L}_0$  is zero. Starting from any  $L \in \mathcal{L}_0$  at  $\tau_0$ , we then compute a tree that is unfolding the time evolution of the TN's vehicle load in  $[\tau_0, \tau_0 + h]$ ; and by merging all

these trees, we derive a node-labelled digraph in form of a forest that we will call the *computation space* (CS) of the prediction (cf. Fig. 5):

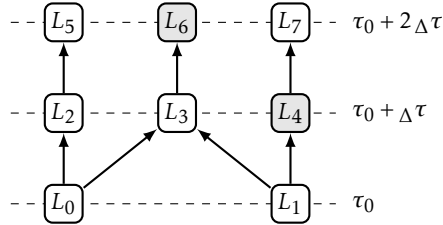


Fig. 5: Unfolding of the time evolution of the TN's vehicle load for the prediction horizon marked with two gray shaded target nodes

The label of every node is a vehicle load, and with every arc connecting one node to another comes along with an increase of  $\Delta\tau$  in the time evolution of the vehicle load. Thus, the label of every node must be a vehicle load in  $\mathcal{L}_0$  if it has no parent. The opposite does not necessarily hold, of course.

Finally, note that the set of all possible jumps from a vehicle load at  $\tau \in \mathcal{H}'$ , with  $\mathcal{H}' := \{\tau_0, \tau_0 + \Delta\tau, \dots, \tau_0 + h - \Delta\tau\}$ , to another vehicle load at  $\tau' := \tau + \Delta\tau$  can be deduced from the specification of the vehicle operation involving the deterministic resolution of all conflicting vehicle movements.

## 5 FORECASTING NETWORK LOADS

By restricting the jumps between the vehicle loads to any discrete point in time  $\tau \in \mathcal{H}$ , we implicitly assume that the continuous dynamics of the SHA can also only change at any  $\tau \in \mathcal{H}$ . In other words, if  $L = (M, \nu_{dr}, \nu_{dw})$  is the vehicle load at  $\tau$ , then it is Eqn. 1 defined by  $M$  that defines all passenger flows in the closed time interval  $[\tau, \tau + \Delta\tau]$ . The emphasis here is on the attribute “closed”: Although, we restrict all jumps to a finite set of consecutive discrete points in time, we are still modelling a continuous-time (stochastic) process; requiring us to ensure the smooth transitioning of all continuous variables. In this context, note that we do not model any continuous-time stochastic process. In fact, Eqn. 1 is Markovian by nature, and its integration into a framework with jumps between vehicles loads gives a continuous-time Markov process again. This is true since the jump condition from a vehicle load at any time instant  $\tau' \in \mathcal{H}'$  to another vehicle load at time instant  $\tau' + \Delta\tau$  is completely defined by the vehicle load at any time instant  $\tau'' \in [\tau - \Delta\tau, \tau]$  together with the passenger loads of all GPs and vehicles. We thus can compute the prediction of the network load starting from some initial distribution at  $\tau_0$  according to the informally described algorithm:

1. Mark the set of all target nodes in the CS.

2. Invert the CS, and compute every path that is starting from a target node and reaching a node at  $\tau_0$ .
3. Invert all computed paths, compute the intersection of their union with the marked CS, and call it the *reduced CS*.
4. Inscribe every node at  $\tau_0$  of the reduced CS with the probability of presence of its vehicle load.
5. Numerically solve the IVP in form of Eqn. 4, Eqn. 5, and Eqn. 8 up to  $\tau + \Delta\tau$  for every node at  $\tau$  and every  $s \in \mathcal{S}$ ; given the PDFs of all passenger loads at  $\tau$  as input.
6. Extract the probabilities of all jumps that can occur at  $\tau + \Delta\tau$  from the computed PDFs at  $\tau + \Delta\tau$ .

Note that the PDFs in point 5) are assumed to be known for  $\tau = \tau_0$ . On the contrary, the PDF for the IVP for a vehicle load  $L$  at  $\tau > \tau_0$  is computed as the sum of all PDFs, that are coming along with a jump to  $L$  multiplied with the probability of that jump. Moreover, note that the probabilities of all jumps leaving a vehicle load in the reduced CS do not necessarily have to sum up to one as opposed to the jumps in the original CS.

As an example, Fig. 6 depicts a reduced CS with two gray shaded target nodes, that was derived from Fig. 5: From  $L_0$  at  $\tau_0$ , we do only have to account for the jump to  $L_3$  at  $\tau_0 + \Delta\tau$  with a probability of 0.1. From  $L_1$  at  $\tau_0$ , however, two jumps at  $\tau_0 + \Delta\tau$  can occur, namely one to  $L_3$  with a probability of 0.2, and another one to  $L_4$  with a probability of 0.8. Thus, in the time interval  $[\tau_0 + \Delta\tau, \tau_0 + 2\Delta\tau]$  the probability of the vehicle load to be  $L_4$  is  $0.6 \times 0.8 = 0.48$ , and to be  $L_3$  is  $0.4 \times 0.1 + 0.6 \times 0.2 = 0.16$ . Then, from reasoning, the probability to be in  $L_6$  at  $\tau + 2\Delta\tau$  must also be 0.16 since the TN's load is evolving continuously in time; requiring that the jump from  $L_3$  to  $L_6$  at  $\tau_0 + 2\Delta\tau$  occurs with probability one.

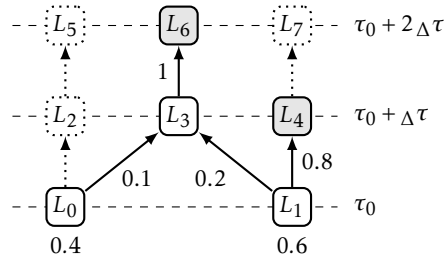


Fig. 6: Reduced CS inscribed with all jump probabilities leading to the set of target nodes (solid lines), together with those nodes and jumps (dotted lines) that belong to the original CS but can be disregarded

## 6 DISCUSSIONS

In this paper we have extended the DHA from [1] in that we have replaced all deterministic passenger arrival processes by their stochastic counterparts for a TN at hand. We then have provided a computation strategy for the forecast of the TN's load in form of a forward reachability problem subjected to the automaton's Markovian hybrid dynamics: Starting from an initial estimation of the network load in form of some PDFs (one for every station), the strategy requires to propagate these PDFs into a target set of timed vehicle loads (= discrete modes). On the way to this target set, the propagation of the PDFs encounters jumps between the discrete modes at a priori defined equidistantly spaced discrete points in time. Within a discrete mode, the propagation is computed by numerically integrating systems of PDEs subjected to boundary conditions. Now the integration of the PDEs is known to be a difficult task, and different strategies have been developed during the last couple of years; with some approaches avoiding it at all such as the grid-based asymptotic approximation of the SDEs by Markov chains from [17]. However, at the same time sophisticated numerical integration methods have been evolved with the multigrid method [16] employed in our ongoing implementation. In this context, we do also consider further simplifications targeting the dimensions of the SDEs, e.g. by aggregating trip profiles with common last "miles", or by refining the discrete modes. Finally, note that - due to space limitations - we could not provide the equations that formally describe the propagation of the densities in the vehicle load tree. We intend to provide them in a future publication, though.

## References

1. S. Haar and S. Theissing, "A hybrid-dynamical model for passenger-flow in transportation systems," in *Proceedings of the 5th IFAC Conference on Analysis and Design of Hybrid Systems*, 2015.
2. R. David and H. Alla, *Discrete, Continuous, and Hybrid Petri Nets*. Springer Publishing Company, 2010.
3. J. D. Schmöcker and W. Adeney, "Metro service delay recovery: comparison of strategies and constraints across systems," *Journal of the Transportation Research Board*, 2005.
4. K. S. Trivedi and V. G. Kulkarni, "FSPNs: Fluid stochastic Petri nets," in *Proceedings of the 14th International Conference on the Application and Theory of Petri nets*, 1993.
5. G. Balbo, "Introduction to generalized stochastic Petri nets," in *Formal Methods for Performance Evaluation*, 2007.
6. G. Horton, V. G. Kulkarni, D. M. Nicol, and K. S. Trivedi, "Fluid stochastic Petri nets: Theory, applications, and solution techniques," *European Journal of Operational Research*, 1998.
7. K. Wolter, "Modelling hybrid systems with fluid stochastic Petri nets," in *Proceedings of the 4th International Conference on Automation of Mixed Processes: Hybrid Dynamic Systems*, 2000.
8. J. Hu, J. Lygeros, and S. Sastry, "Towards a theory of stochastic hybrid systems," in *Hybrid Systems: Computation and Control*. Springer Berlin Heidelberg, 2000.

9. M. L. Bujorianu, J. Lygeros, and C. B. Bujorianu, "Toward a general theory of stochastic hybrid systems," in *Stochastic Hybrid Systems*. Springer Berlin Heidelberg, 2006.
10. J. Bect, "A unifying formulation of the Fokker-Planck-Kolmogorov equation for general stochastic hybrid systems," *ArXiv e-prints*, 2008.
11. M. Prandini and J. Hu, "A numerical approximation scheme for reachability analysis of stochastic hybrid systems with state-dependent switchings," in *Proceedings of the 46th IEEE Conference on Decision and Control*, 2007.
12. T. Szabados, "An elementary introduction to the Wiener process and stochastic integrals," *ArXiv e-prints*, 2010.
13. D. J. C. MacKay, "Introduction to monte carlo methods," in *Proceedings of the NATO Advanced Study Institute on Learning in Graphical Models*, 1998.
14. W. Press, W. Vetterling, S. Teukolsky, and B. Flannery, *Numerical recipes in C - The art of scientific computing*, 2nd ed. Cambridge University Press, 1992.
15. C. Gardiner, *Handbook of Stochastic Methods*, 2nd ed. Springer Publishing Company, 1985.
16. H. b. Zubair, C. W. Oosterlee, and R. Wienands, "Multigrid for high-dimensional elliptic partial differential equations on non-equidistant grids," *SIAM J. Sci. Comput.*, 2007.
17. M. Prandini and J. Hu, "A stochastic approximation method for reachability computations," in *Stochastic Hybrid Systems*, 2006.