Stochastic Delay Prediction in Large Train Networks

Anabell Berger, Andreas Gebhardt, Matthias Müller-Hannemann, and Martin Ostrowski
Stochastic Delay Prediction in Large Train Networks

Annabell Berger, Andreas Gebhardt, Matthias Müller-Hannemann, and Martin Ostrowski
In daily operation, railway traffic always deviates from the planned schedule to a certain extent. Primary initial delays of trains may cause a whole cascade of secondary delays of other trains over the entire network. In this paper, we propose a stochastic model for delay propagation and forecasts of arrival and departure events which is applicable to all kind of public transport (not only to railway traffic). Our model is fully realistic, it includes general waiting policies (how long do trains wait for delayed feeder trains), it uses driving time profiles (discrete distributions) on travel arcs which depend on the departure time, and it incorporates the catch-up potential of buffer times on driving sections and trains stops. The model is suited for an online scenario where a massive stream of update messages on the current status of trains arrives which has to be propagated through the whole network.

Efficient stochastic propagation of delays and forecasts for the likelihood that planned transfers between connecting trains are feasible have important applications in online timetable information, in delay management and train disposition, and in stability analysis of timetables.

The proposed approach has been implemented and evaluated on the German timetable of 2011 with waiting policies of Deutsche Bahn AG. A complete stochastic delay propagation for the whole German train network and a whole day can be performed in less than 14 seconds on a PC. We tested our propagation algorithm with artificial discrete travel time distributions which can be parameterized by the size of their fluctuations. Our forecast are compared with real data for different types of days, namely a weekday and a day on the weekend.

As can be expected, the range of the support of the distributions increases with the size of fluctuations and an increase of the time horizon, but all scenarios turned out to be computable in almost real time. Hence, stochastic simulation of delays is efficient enough to be applicable in practice, but the forecast quality requires further adjustments of our artificial travel time distributions to estimations from real data.

**Keywords**: stochastic delay propagation, timetable information, delay management, train disposition, stability analysis

## 1 Introduction

**Motivation.** Train delays occur for various reasons: Disruptions in the operations flow, accidents, malfunctioning or damaged equipment, construction work, repair work, and extreme weather conditions like snow and ice, floods, and landslides, to name just a few. Initial delays of these types are called primary delays. They usually induce a whole cascade of secondary delays of other trains which have to wait according to certain waiting policies between connecting trains. On a typical day of operation of German Railways, an online system has to handle about 6 million forecast messages about (mostly small) changes with respect to the planned schedule and the latest prediction of the current situation. As a consequence of disruptions, we have to handle dynamically changing schedules. Thus, a graph model representing the current schedule has to be updated at a rate of about 70 update operations per second [13]. These update operations include is-messages, forecast-messages, train cancellations, track changes etc.

Delay cascades cannot be forecast exactly due to several stochastic influences. For example, trains can drive faster than planned or stay shorter at stations as scheduled and so catch up some of their delay. In fact, to make the schedule more robust, certain slacks are usually integrated
into the planned schedule. For example, the planned driving time on a track segment is increased by a certain percentage over the minimum driving time when the train operates with full speed. Likewise, trains have larger staying times at stations when necessary for boarding and deboarding of passengers.

Stochastic forecasts can be used for several purposes:

1. **Ontrip timetable information:** The arrival and departure time distribution can be used to evaluate the reliability of a planned transfer and then used in a multi-criteria setting as an additional objective.

2. **Delay management and train disposition:** Dispatchers have to decide whether a train should wait for another delayed train. These decisions are quite complex, and so it is very helpful to evaluate the reliability of forecasts of arrival and departure times as a decision aid. This information can be used for explicit human decisions or in an automatic disposition system which tries to find globally optimal waiting decisions.

3. **“Stability analysis” of the planned schedule:** Stochastic simulations of delays allow for a quantitative evaluation how small delays (due to small fluctuations) propagate through the system? They help to study the robustness of the schedule.

**Related Work.** Efficient deterministic propagation of primary and secondary delays has been done by Frede et al. [5] and Müller-Hannemann and Schnee [13]. They demonstrated that even massive delay data streams can be propagated instantly, making this approach feasible for real-time multi-criteria timetable information. Goverde [10] recently presented an efficient deterministic delay propagation algorithm for periodic timetables.

Train event networks are very similar to project networks. In stochastic project networks (PERT-networks), the nodes are project events and arcs correspond to activities. If the duration of each activity has an associated probability distribution, one is typically interested in critical paths or in the distribution function of the overall project completion time. The computation of the distribution function is computationally very hard, even evaluating this function at a single point is \#P-complete in general [11].

Stochastic models for the propagation of delays have been studied intensively, most importantly by Carey and Kwieciński [3, 4] and Meester and Muns [12]. They propose to use approximations of delay distributions to reduce the computational effort and study the error propagation for such approximations. However, they do not model waiting policies for connecting trains. For the computation of propagated delays, the distributions are treated as if they were independent. Although it is difficult to bound the consequences of the independence assumption quantitatively, Meester and Muns argue that the effect of their approximations is small. The experimental evaluation of [12] has been conducted on a “toy network” of ten stations and four train lines. A similar approach has been taken by Büker [2]. Compared to our work, his experiments are only based on relatively small subnetworks. Stability analysis of railway timetables using a max-plus algebra approach has been done by Goverde [9]. See also the PhD-thesis of Yuan [15] for further references and an in-depth discussion of models. He performs an extensive statistical analysis of train delays and delay propagation in some part of the Netherlands.

A related field of current research is disposition and delay management. Gatto et al. [6, 7] have studied the complexity of delay management for different scenarios and have developed efficient algorithms for certain special cases using dynamic programming and minimum cut computations. Various policies for delays have been discussed, for example by Ginkel and Schöbel [8]. Schöbel [14] also proposed integer programming models for delay management. These models work with static forecasts of delays. Policies for delays in a stochastic context are treated in [1].

**Our Contribution.** We present in the following section a concise and realistic stochastic model for delay propagation and calculation of arrival and departure time distributions in public transport. The model is formulated with respect to an event graph which models the train schedule and
Stochastic Delay Prediction in Large Train Networks

the waiting conditions between planned transfer possibilities. In the event graph, each departure and arrival event corresponds to a vertex, and vertices are linked by two types of arcs: travel arcs and transfer arcs. A travel arc connects a departure with the very next arrival event of a train. Transfer arcs either model the stop of a train at some station and so link the corresponding arrival and departure events, or they connect the arrival event of some train \( A \) to the departure event of some other train \( B \) at the same station in case of a planned transfer.

We propose a fully realistic stochastic model. It includes general waiting policies (how long do trains wait for delayed feeder trains), it uses driving time profiles (discrete distributions) on travel arcs which depend on the departure time, but also on train category or track conditions. Moreover, our model incorporates the catch-up potential of buffer times on driving sections and at trains stops. We believe that the resulting model is quite elegant which made it possible to implement it with a reasonable effort.

Distributions of travel times on travel arcs can be chosen arbitrarily which allows to test very different scenarios, in particular to stress the system to its limits. However, there is a rich body of statistical data about delays and the potential to reduce them in the past (detailed for train categories and track segments). Realistic assumptions on delay distributions on individual track segments should be based on them.

A crucial property of our approach is that it allows dynamic updates with respect to new delay messages. Given incoming messages (new delays or updates of existing delays, and current effective status messages of trains) from some external source, we immediately propagate these messages through the whole network. The event graph is a directed acyclic graph. Therefore, delay propagation can be done in topological order of events. For start up it is necessary to propagate once the initial distributions over the whole event graph. Afterwards new forecasts and effective status messages are only propagated in the forward cone of the corresponding event, i.e. in the part of the network which can be reached from it.

We work with two types of distributions: 1-point distributions of already realized events and arbitrary discrete distributions of events which still lie in the future.

Although stochastic delay propagation is computationally quite expensive, we managed to implement a version which is fast enough to be used in an online system.

Experiments with a prototypical implementation on the whole German train network and realistic waiting rules between connecting trains require less than 14s to propagate all discrete distributions for a full traffic day. Simulations with several distributions of travel times on travel arcs yield interesting insights into the robustness of the planned schedule against small fluctuations.

We compare our predictions with realized event times for an ordinary and a disastrous day to analyse the quality of our predictions. In addition, we determine the width of the supports of the computed distributions. By that, we assess the precision of our forecasts and how they depend on the difference between the prediction time and the actual time. We perform experiments with four different set of waiting rules between connecting trains.

**Overview.** In the following section we describe in detail the event graph and our stochastic model and its underlying assumptions. Afterwards we discuss in detail, how arrival and departure probabilities can be computed for all events. In Section 4, we report first experimental results with a prototypical implementation.

2 The Stochastic Model

**The timetable and its corresponding event graph.** A time table \( TT := (P, S, C) \) consists of a tuple of sets. Let \( P \) be the set of trains, \( S \) the set of stations and \( C \) the set of elementary connections, that is

\[
C := \left\{ c = (p, s, s', t_d, t_a) \mid \text{train } p \in P \text{ leaves station } s \text{ at time } t_d, \text{ The next stop of } p \text{ is at station } s' \text{ at time } t_a \right\}.
\]
A time table $TT$ is valid for a number of $N$ traffic days. A validity function $v: Z \mapsto \{0, 1\}^N$ determines on which traffic days the train operates. We denote a time value $t$ in $TT$ by
\[ t := \alpha \cdot 1440 + \beta \text{ with } \alpha \in [0, N), \beta \in [0, 1439]. \]
The actual time within a day is then $t_{\text{day}} = t \mod 1440$ and the actual day $d = \lfloor \frac{t}{1440} \rfloor$.

We define with respect to the set of elementary connections $C$ a set of departure events $\text{Dep}$ and arrival events $\text{Arr}$ for all $v \in V$. Each event $\text{dep}_v := (\text{time}, \text{train}) \in \text{Dep}$ and $\text{arr}_v := (\text{time}, \text{train}) \in \text{Arr}$ represents exactly one departure or arrival event which consists of the two attributes $\text{time}$ and $\text{train}$. Staying times at a station $v$ can be lower and upper bounded by minimum and maximum staying times $\text{minstay}(\text{arr}_v, \text{dep}_v), \text{maxstay}(\text{arr}_v, \text{dep}_v) \in \mathbb{Z}^+$ which have to be respected between different events in $v$. Staying times ensure the possibility to transfer from one train (the so-called feeder train) to the next. We denote by $G := (V, A)$ the event graph with $V := \text{Dep} \cup \text{Arr}$ and the arc set $A := A_{\text{travel}} \cup A_{\text{transfer}}$ consisting of the travel arc set $A_{\text{travel}} := \{(\text{dep}_w, \text{arr}_w)| \text{ there exists } c \in C \text{ with } t_d = \text{dep}_v(\text{time}), t_a = \text{arr}_w(\text{time}), v = s, w = s' \wedge p = \text{dep}_v(\text{train}) = \text{arr}_w(\text{train})\}$
and the transfer arc set $A_{\text{transfer}} := \{\text{arr}_v, \text{dep}_v)| \text{ arr}_v \in \text{Arr}, \text{dep}_v \in \text{Dep}, \text{minstay}(\text{arr}_v, \text{dep}_v) \leq \text{dep}_v(\text{time}) - \text{arr}_v(\text{time}) \leq \text{maxstay}(\text{arr}_v, \text{dep}_v)\}$.

Furthermore, we define waiting times $\text{wait}_\text{transfer} : A_{\text{transfer}} \mapsto \mathbb{Z}^+ \cup \{\infty\}$ where we denote by $\text{wait}_\text{transfer}(\text{arr}_v, \text{dep}_v)$ the number of time units which $\text{train}(\text{dep}_v)$ may depart later as planned time $\text{time}(\text{dep}_v)$ with respect to its feeder train $\text{train}(\text{arr}_v)$. Clearly, $\text{wait}_\text{transfer}(\text{arr}_v, \text{dep}_v) = \infty$ if $\text{train}(\text{arr}_v) = \text{train}(\text{dep}_v)$, because a train cannot depart before its arrival. We define a further waiting time $\text{wait} : \text{Dep} \mapsto \mathbb{Z}^+$ with $\text{wait}(\text{dep}_v) := \max \{\text{wait}_\text{transfer}(\text{arr}_v, \text{dep}_v)| (\text{arr}_v, \text{dep}_v) \in A_{\text{transfer}} \wedge \text{train}(\text{arr}_v) \neq \text{train}(\text{dep}_v)\}$.

If some train is delayed by more than $\text{wait}(\text{dep}_v)$, then its departure time depends on no other train irrespectively of their delays. Each travel arc $(\text{dep}_v, \text{arr}_w) \in A_{\text{travel}}$ possesses a scheduled travel time $\text{arr}_w(\text{time}) - \text{dep}_v(\text{time})$ and a minimum possible travel time $\text{mintt}(\text{dep}_v, \text{arr}_w) \in \mathbb{Z}^+$ with $\text{mintt}(\text{dep}_v, \text{arr}_w) \leq \text{arr}_w(\text{time}) - \text{dep}_v(\text{time})$. If train $\text{train}(\text{dep}_v)$ departs too late at $v$ their exists the possibility to regain the lost time. We define a realization time $t_e(\text{event})$ for each event and call the current time point $\text{update time} t_{\text{update}}$. Note that scheduled time points (see our attributes in departure or arrival events) are denoted as ‘time’.

**Model assumptions.** In the following, we specify and discuss our model assumptions. The general scenario is that we obtain a stream of online messages about the delay status of trains (so-called status messages) from the railway company, i.e., for each train, the difference between the scheduled and the realization time for departure and arrival events is measured and reported.

**Assumption 1.** With respect to status messages, a train can arrive or depart at any time after the planned arrival or departure time, respectively.

Of course, a train shall never depart before its scheduled departure time. In reality, a train may arrive somewhat early, but then its waiting time at the station will be increased. Thus our model assumption does not make a difference for delay propagation, but simplifies the mathematical model.

For compatibility to Assumption 1, we demand the following.

**Assumption 2.** With respect to our forecasts of arrival and departure time distributions, no train departs before its scheduled time or arrives at a station before its planned arrival time.
3 Departure and Arrival Probabilities

3.1 Travel Time, Departure and Arrival Random Variables

Let $(\Omega, \mathcal{A}, P)$ be a discrete probability space with sample space $\Omega$, $\sigma$-algebra $\mathcal{A}$ and probability measure $P$. Furthermore, let $T \subset \mathbb{Z}^+$ a discrete set of time points. We define with respect to a current time $t_{update}$ for each event $event \in Dep \cup Arr$ a discrete random variable

$$X_{event} : \Omega \mapsto \{event(time), event(time) + 1, \ldots\}.$$

We call a variable departure random variable if $event = dep_v$ and arrival random variable for $event = arr_v$, where $dep_v, arr_v \in Dep \cup Arr$. The range of $X_{event}(\Omega)$ is $\{time(event), time(event) + 1, \ldots\}$ by our Assumption 2. With respect to Assumption 3 we state that all arrival random variables $X_{arr_v}$ with $arr_v \in Arr$ for a single station $v \in S$ are pairwise stochastically independent with respect to probability measure $P$. This means that for all pairs $(t, t') \in \{arr_v(time), \ldots\} \times \{arr_v'(time), \ldots\}$ it follows that

$$P(X_{arr_v}^{-1}(\{t\}) \cap X_{arr_v'}^{-1}(\{t'\})) = P(X_{arr_v}^{-1}(\{t\})) \cdot P(X_{arr_v'}^{-1}(\{t'\})).$$

Furthermore, we distinguish between realized and not realized random variables. For all realized events (in such cases $t_v(event) \leq t_{update}$) we state $P(X_{event}^{-1}(t_v(event))) := 1$. Non-realized events are in general not ‘one-point-distributed’.

Note, that an event can possess an infinite set of different random variables depending on the current time point and several possibilities for realization times at each event. Here, we only describe one ‘effective situation’ for exactly one time point $t_{update}$ for simplicity in our notation.

In such a static situation we are able to determine the current random variable deterministically. Furthermore, we need a random variable which describes possible travel times on each arc $(dep_v, arr_w) \in A_{travel}$. Generally, we want to model the case that a train can regain the lost time with a smaller travel time as the planned travel time $arr_w(time) - dep_w(time)$. Assumption 2 ensures that we may not arrive at an earlier time as $arr_w(time)$. Hence, we need for each arc $(dep_v, arr_w) \in A_{travel}$ a sequence of discrete travel time variables $X^{t}_{(dep_v, arr_w)}(t) \in TP$ for each possible departure time point $t \in TP := \{dep_v(time), dep_v(time) + 1, \ldots\}$ with

$$X^{t}_{(dep_v, arr_w)} : \Omega \mapsto \{mintt(dep_v, arr_w), \ldots, arr_w(time) - dep_v(time) + k\}. $$
A. Berger, A. Gebhardt, M. Müller-Hannemann, and M. Ostrowski

Figure 1: Possible travel times on an arc \((\text{dep}_v, \text{arr}_w)\) depending on the actual departure time. We use the abbreviations \(t := \text{dep}_v(\text{time}), d := \text{arr}_w(\text{time}) - \text{mintt}(\text{dep}_v, \text{arr}_w), \text{min} := \text{mintt}(\text{dep}_v, \text{arr}_w)\) and scheduled:= \(\text{arr}_w(\text{time}) - \text{dep}_v(\text{time})\). The allowed fluctuation above the scheduled travel time is here chosen as \(k = 2\). The data points connected by lines represent all travel times which lead to the same arrival time at \(w\). The points in the same column \(t\) correspond to all possible travel times for a fixed distribution \(X_t(\text{dep}_v, \text{arr}_w)\).

To model small fluctuations of travel times on standard operation days, we use parameter \(k \in \mathbb{N}\) specifying the width of the support of these distributions. Hence, in our experiments we will choose a fairly small value for \(k\). To satisfy Assumption 2 of our model, we have to distinguish random variables for different times with respect to their time distance to the scheduled arrival times. That means that the probability for a time \(t\) must be zero if the distance between a forecasted time and the scheduled arrival time \(\text{arr}_w(\text{time})\) is more than \(\text{mintt}(\text{dep}_v, \text{arr}_w)\). We set \(P((X^t_t(\text{dep}_v, \text{arr}_w))^{-1}(\{\text{mintt}(\text{dep}_v, \text{arr}_w), \ldots, \text{arr}_w(\text{time}) - t - 1\})) := 0\) for all \(t \in \{\text{dep}_v(\text{time}), \ldots, d\}\) because our Assumption 2 prohibits to arrive earlier as planned. Clearly, it is necessary to model for all these points in time \(t\) distinct random variables \(X_t(\text{dep}_v, \text{arr}_w)\). In theory, we are able to distinguish infinite many of such random variables. In our experiments, we restrict ourselves to the case where all random variables are identically from a certain point of time \(d\) onwards. We set \(d := \text{arr}_w(\text{time}) - \text{mintt}(\text{dep}_v, \text{arr}_w) - 1\) and define

\[X_t^{d+1}(\text{dep}_v, \text{arr}_w) := X_{d+1}(\text{dep}_v, \text{arr}_w)\]

for all \(t > d\).

We make this assumption because we have no data for heavy delays. Consider Figure 1 for an example of travel time distributions.

3.2 Departure Random Variables and Departure Probabilities

When we reach the current time point \(t_{\text{update}}\), we replace for all events with a realization time \(t_r(\text{event}) \leq t_{\text{update}}\) their discrete departure or arrival random variables with the above defined ‘one point distribution’ (if the data is available). Afterwards, we can compute — following a topological ordering of the acyclic event graph — all succeeding random variables. In a next step we want to describe how one can compute the departure random variable for an event which has not yet been realized. Fig. 2 gives a schematic view of the scenario where a departure event depends on several feeding trains.

For the determination of a departure random variable we distinguish between three cases.

1. Train \(\text{train}(\text{dep}_v)\) departs at its scheduled time \(\text{dep}_v(\text{time})\).
2. Train \( \text{train}(\text{dep}_v) \) departs at \( t \in \{ \text{dep}_v(\text{time}) + 1, \ldots, \text{dep}_v(\text{time}) + \text{wait}(\text{dep}_v) \} \).

3. Train \( \text{train}(\text{dep}_v) \) departs at \( t \in \{ \text{dep}_v(\text{time}) + \text{wait}(\text{dep}_v) + 1, \ldots \} \).

Let us denote the set of all arrival events of feeder trains by \( F := \{ \text{arr}_i^v | (\text{arr}_i^v, \text{dep}_v) \in A_{\text{transfer}}, \text{train}(\text{arr}_i^v) \neq \text{train}(\text{dep}_v) \} \).

In the example of Figure 2, all three cases occur, namely Case 1 for \( \text{dep}_v(\text{time}) = 10:03 \), Case 2 for \( \text{dep}_v(\text{time}) = 10:04, 10:05 \) (because \( \text{wait}(\text{dep}_v) = 2 \)), and Case 3 for \( \text{dep}_v(\text{time}) = 10:06 \).

Case (1) occurs if train \( \text{train}(\text{arr}_v) \) arrives at

\[
I_i(t) := \{ \text{arr}_i^v(\text{time}), \ldots, t - \text{min stay}(\text{arr}_i^v, \text{dep}_v) - 1 \} \cup \\
\{ \text{dep}_v(\text{time}) - \text{min stay}(\text{arr}_i^v, \text{dep}_v) + \text{wait}(\text{arr}_i^v, \text{dep}_v) + 1, \ldots \}
\]

and \( I_i(t) := \{ \text{arr}_v(\text{time}), \ldots, t - \text{min stay}(\text{arr}_v, \text{dep}_v) - 1 \} \).

Furthermore, we need slightly different index sets

\[
J_i(t) := \{ \text{arr}_i^v(\text{time}), \ldots, t - \text{min stay}(\text{arr}_i^v, \text{dep}_v) \} \cup \\
\{ \text{dep}_v(\text{time}) - \text{min stay}(\text{arr}_i^v, \text{dep}_v) + \text{wait}_{\text{transfer}}(\text{arr}_i^v, \text{dep}_v) + 1, \ldots \}
\]

and \( J_i(t) := \{ \text{arr}_v(\text{time}), \ldots, t - \text{min stay}(\text{arr}_v, \text{dep}_v) \} \). Finally, we denote

\[
m_i := \text{min stay}(\text{arr}_i^v, \text{dep}_v).
\]

For case (1) we compute the preimages of a departure random variable with

\[
X_{\text{dep}_v}^{-1}(\{\text{dep}_v(\text{time})\}) = \bigcap_{i=1}^l X_{\text{arr}_i^v}^{-1}(J_i(\text{dep}_v(\text{time}))).
\]

By Assumption 3, and if we denote \( p_{\text{arr}_i^v}(t) := P(X_{\text{arr}_i^v}^{-1}(\{t\})) \) and \( p_{\text{dep}_v}(t) := P(X_{\text{dep}_v}^{-1}(\{t\})) \) we get

\[
p_{\text{dep}_v}(\text{dep}_v(\text{time})) = \prod_{i=1}^\lambda \sum_{\lambda \in J_i(\text{dep}_v(\text{time}))} p_{\text{arr}_i^v}(\lambda).
\]

We call \( p_{\text{dep}_v} \) the departure probability and \( p_{\text{arr}_v} \) the arrival probability. For our example see Figure 2 we get for a departure time of 10:03 the probability

\[
p_{\text{dep}_v}(10:03) = p_{\text{arr}_1^v}(10:01) \cdot p_{\text{arr}_2^v}(10:02) \cdot p_{\text{arr}_3^v}(\{10:01, 10:02\} \cup \{10:05\})
\]

\[
= 0.5 \cdot 0.2 \cdot (0.4 + 0.2 + 0.1)
\]

\[
= 0.07.
\]

Case (2) happens if train \( \text{train}(\text{arr}_v) \) arrives in interval \( \{ \text{arr}_v(\text{time}), \ldots, t - \text{min stay}(\text{arr}_v, \text{dep}_v) \} \) and at least one feeder train \( \text{train}(\text{arr}_w^v) \) with \( \text{arr}_w^v \in F \) arrives exactly at time point \( t -
minstay(arr^{v_0}, dep_v). We define with respect to possible departure times \( t \in \{dep_v(time) + 1, \ldots, dep_v(time) + wait(dep_v)\} \) the set of all ‘exact’ time point tuples as

\[
A_t := \{(t_1, \ldots, t_l) | (t_1, \ldots, t_l) \in (\times_{i=1}^l J_i(t)) \wedge \text{there exists } i_0 < l \text{ with } t_{i_0} = t - m_{i_0}\}.
\]

For a departure random variable in case (2) we get

\[
X^{-1}_{dep_v}(\{t\}) = \bigcup_{(t_1, \ldots, t_l) \in A_t} \left( \bigcap_{i=1}^l X^{-1}_{arr_{v_i}}(\{t_i\}) \right).
\]

This formulation is compact but we have to consider exponentially many disjoint subsets of \( \Omega \) leading to a non-efficient algorithm for computing \( X_{dep_v} \). Instead, we rearrange these preimages by applying the well-known ‘De Morgan-rules’ such that we get only polynomially many disjoint subsets of \( \Omega \). With this setting we get

\[
X^{-1}_{dep_v}(\{t\}) = \bigcup_{j=0}^{l-1} \bigcap_{i=1}^j X^{-1}_{arr_{v_i}}(I_i(t)) \bigcap \{X^{-1}_{arr_{v_{j+1}}}(\{t - m_{j+1}\}) \bigcap \bigcup_{j=0}^{l-1} S_j \}
\]

Using \( \sigma \)-additivity to compute the elementary probabilities \( p_{dep_v}(t) := P(X^{-1}_{dep_v}(\{t\})) \) we have to show that for all pairs \( j, j' \in \{0, \ldots, k-1\} \) the sets \( S_j, S_{j'} \) are disjoint. It is sufficient to prove that for an arbitrary \( j_0 \) the sets \( S_{j_0} \) and \( S_{j_0+1} \) are pairwise disjoint. Assume there exists an \( \omega \in \Omega \) with \( \omega \in S_{j_0} \cap S_{j_0+1} \). Then it follows that \( X_{arr_{v_{j_0+1}}}(\omega) = t - m_{j_0+1} \) and \( X_{arr_{v_{j_0+1}}}(\omega) \in I_{j_0+1} \). Because \( t - m_{j_0+1} \notin I_{j_0+1} \) this is a contradiction. Hence, we can apply \( \sigma \)-additivity and use Assumption 3 that our random variables are stochastically independent. For case (2) we obtain

\[
p_{dep_v}(t) = \sum_{j=0}^{l-1} \prod_{i=1}^j \left( \sum_{\lambda \in I_i(t)} p_{arr_{v_i}}(\lambda) \right) \cdot p_{arr_{v_{j+1}}}(t - m_{j+1}) \cdot \prod_{i=j+2}^l \left( \sum_{\lambda \in J_i(t)} p_{arr_{v_i}}(\lambda) \right).
\]

Back to our example (see Figure 2) we get for a departure time of 10:03 the departure probability
and for departure time 10:04 the departure probability

\[
p_{\text{dep}, 10:04} = p_{\text{arr}}(10:02) \cdot p_{\text{arr}}^2(\{10:02, 10:03\}) \cdot p_{\text{arr}}^3(\{10:01, 10:02, 10:03\} \cup \{10:05\}) + p_{\text{arr}}^3(10:01) \cdot p_{\text{arr}}^2(10:03) \cdot p_{\text{arr}}^3(\{10:01, 10:02, 10:03\} \cup \{10:05\}) + p_{\text{arr}}^3(10:01) \cdot p_{\text{arr}}^2(10:02) \cdot p_{\text{arr}}^3(10:03) = 0.2 \cdot 0.7 \cdot 0.9 + 0.5 \cdot 0.5 \cdot 0.9 + 0.5 \cdot 0.2 \cdot 0.2 = 0.371.
\]

and for departure time 10:05 the departure probability

\[
p_{\text{dep}, 10:05} = p_{\text{arr}}(10:03) \cdot p_{\text{arr}}(\{10:02, 10:04\}) \cdot p_{\text{arr}}^3(\{10:01, \ldots, 10:05\}) + p_{\text{arr}}^2(\{10:01, 10:02\}) \cdot p_{\text{arr}}^2(10:04) \cdot p_{\text{arr}}^3(\{10:01, \ldots, 10:05\}) + p_{\text{arr}}^2(10:01, 10:02) \cdot p_{\text{arr}}^2(\{10:02, 10:03\}) \cdot p_{\text{arr}}^2(10:04) = 0.2 \cdot 1 + 0.7 \cdot 0.3 \cdot 1 + 0.7 \cdot 0.7 \cdot 0.1 = 0.459.
\]

Case (3) is much simpler because the departure time of train train(dep) only depends on its arriving time arr(time). That is

\[
X_{\text{dep}}^{-1}(\{t\}) = X_{\text{arr}}^{-1}(\{t - m\})
\]

and results in

\[
p_{\text{dep}, t} = p_{\text{arr}}(t - m).
\]

In our example (see again Figure 2) we get for a departure time of 10:06 the departure probability \(p_{\text{dep}, 10:06} = p_{\text{arr}}(10:04) = 0.1\).

The case that train train(dep) starts at station \(v \in S\) is also simpler. Obviously, its departure time only depends on feeder trains. We can take all above computations but ignore the arrival event \(arr^k\) in case (1) and case (2). For case (3) we set \(p_{\text{dep}, t} := 0\) for all \(t \in \{\text{dep}(\text{time}) + \text{wait}_{\text{transfer}}(\text{dep}) + 1, \ldots\}\).

### 3.3 Arrival Random Variables and Arrival Probabilities

Let \((\text{dep}, \text{arr}_w) \in A_{\text{travel}}\) be a travel arc. The arriving time on \(w\) depends on the departure time at \(v \in S\) and all possible travel times on this travel arc at this time. We denote the possible travel time set with \(\text{PTT} := \{\text{mintt}(\text{dep}, \text{arr}_w), \ldots, \text{arr}_w(\text{time}) - \text{dep}_v(\text{time}) + k\} \) with \(k \in \mathbb{N}\). Formally, we get for each \(t \in \{\text{arr}_w(\text{time}), \ldots\}\)

\[
X_{\text{arr}}^{-1}(\{t\}) = \bigcup_{j \in \text{PTT}} (X_{\text{dep}}^{-1}(\{t - j\})) \cap (X_{\text{(dep, arr}_w\text{)}}^{-1}(\{j\})).
\]

We can apply \(\sigma\)-additivity for probability measure \(P\). We set

\[
p_{\text{arr}_w}(t) := P(X_{\text{(dep, arr}_w\text{)}}^{-1}(\{t\}))
\]

and get the arrival probability for an arrival event \(\text{arr}_w\) at time \(t\) as

\[
p_{\text{arr}_w}(t) = \sum_{j \in \text{PTT}} p_{\text{dep}, t - j} \cdot p_{\text{(dep, arr}_w\text{)}}^{-1}(j).
\]

### 4 Experiments

In this section, we present the results of several experiments. These have been guided by the following questions:
<table>
<thead>
<tr>
<th>Date</th>
<th>Available Event Data</th>
<th>Non-available but Presumably in Due Time Event Data</th>
<th>Non-available Event Data Without Any Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>20.02.2011</td>
<td>45</td>
<td>38</td>
<td>17</td>
</tr>
<tr>
<td>10.03.2011</td>
<td>51</td>
<td>44</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 1: Data availability with respect to all events. Percentage of available and non-available data. A fraction of non-available data can be assumed to be “in due time” (3rd column.)

Figure 3: Comparison of the absolute differences between realized and planned timestamps for two different days, 10.03.2011 (above, dotted: a Thursday) and 20.02.2011 (below: a Sunday).

1. How well do our forecasts meet the observed data? We compare our forecast for arrival and departure time distribution with realized timestamps of real world data from German Railways. More precisely, we use two different weekdays, namely a Thursday and a Sunday. In Table 1, we give an overview about the data situation for both days. This is necessary, because we did not get all realized event times from German Railways. For about 30% of the regional trains we have no information about their realized departure or arrival times. For the 10.03.2011 we have collected 289,459 messages about realized event times and for the 20.02.2011 we got 193,461 messages. Following statements of German Railways, we may assume that the rest of non-available messages were “in due time”, what here means that all these trains had at most 1 minute of delay.

With respect to this data situation, we determined the absolute difference between realized timestamps (of observed real world data) and the planned schedule time to measure the “strength of a travel chaos”, see Figure 3. The results of this experiment build the basis for the interpretation of further experiments which investigate the quality of different waiting rules. Furthermore, we want to validate our forecasts with respect to a given time horizon. We expect to give better forecasts for the near future than to events further apart, but how fast drops the accuracy of our predictions.

2. How precise are our forecasts, i.e. how narrow or wide are the computed distributions? The smaller the support of the distribution, the more meaningful is our forecast. A ‘one-point’-distribution is clearly a much stronger statement than a widely stretched distribution.

How is the robustness of different timetables with respect to distinct waiting rules. In this scenario we do not use delay data. We only concentrate on the question, how well can slack times compensate small delays.
For the questions in (1), we propose two different measures. A first one computes absolute differences of the expectation values for each departure and arrival time distribution. We count the number of events with difference \( x \) for each \( x \in \{0, \ldots, 1439\} \). A second one determines the fraction of distributions such that the realized time is in the interval of a \( \kappa \)-quantile with \( \kappa \in [0, 1] \), containing all time points such that the cumulative distribution function has a value \( \leq \kappa \).

For the questions in (2), we determine the number of distributions for each constant support width. Here, we consider also the planned times to study the enlargements of supports with increasing distance to the current point of time.

**Test instances and environment.**

Our computational study is based on the German train schedule of 2011, with actual data of realized departure and arrival times for days in February and March 2011. Each day of operation has about 300,000 departure and arrival events per day.

All experiments were run on a PC (Intel(R) Xeon(R), 2.93GHz, 4MB cache, 47GB main memory under ubuntu linux version 8.10). Only one core has been used by our program. Our code is written in C++ and has been compiled with g++ 4.4.3 and compile option -O3.

**Delay distributions on travel arcs.**

For our simulation experiments we use two types of distributions: a uniform distribution and a kind of unimodal distribution with a peak at the scheduled travel time (recall our definitions from Subsection 3.1). Our unimodal distribution is parameterized by \( k \) which controls the support size. For fluctuation parameter \( k \) the support has width \( 2k + 1 \), and the distribution assigns the probabilities  \( \frac{1}{2k+1}, \frac{1}{2k+1}, \ldots, \frac{1}{2k+1} \) to the travel times \( \text{mintt}, \text{mintt} + 1, \ldots, s, s + 1, \ldots, s + k \), where \( s \) denotes the scheduled travel time.

We select the travel time distribution of a travel arc depending on the actual departure time \( \text{dep}_v(\text{time}) \). If the departure time at departure event \( \text{dep}_v \) is between the scheduled time \( \text{dep}_v(\text{time}) \) and \( \text{arr}_v(\text{time}) - \text{mintt}(\text{dep}_v, \text{arr}_v) \) we either apply the uniform or the unimodal distribution. If the actual departure time is above \( \text{arr}_v(\text{time}) - \text{mintt}(\text{dep}_v, \text{arr}_v) \), we always apply the unimodal distribution.

**Effect on the whole network.**

The running time for the computation of all arrival and departure distributions of a whole day takes only a few seconds for the whole day. Hence, we are able to determine forecasts in real time. In our experiments we use the fluctuation parameter \( k \in \{1, 2, 3, 4, 5\} \) for travel times, i.e., the maximum permitted additional travel time for each train between two stops. For the waiting times \( \text{wait}_{\text{transfer}} \) we use four different scenarios.

1. **rule-based:** We use the waiting rules from German Railways.
2. **always:** Each train has to wait for all of its feeder trains. Hence, \( \text{wait}_{\text{transfer}}(\text{arr}_v, \text{dep}_v) = \infty \) for all \((\text{arr}_v, \text{dep}_v) \in A_{\text{transfer}}\).
3. **never:** No train has to wait for another train. We get \( \text{wait}_{\text{transfer}}(\text{arr}_v, \text{dep}_v) = 0 \) for all \((\text{arr}_v, \text{dep}_v) \in A_{\text{transfer}}\).
4. **static:** A train has to wait for a feeder train exactly \( x \) minutes. We set \( x := 5 \) and \( \text{wait}_{\text{transfer}}(\text{arr}_v, \text{dep}_v) = 5 \).

The four waiting strategies can be ordered with respect to the number of restrictions they impose: **never** has obviously the fewest number of waiting conditions, it is followed by **rule-based**, and **static**, while **always** is on the other extreme.

Table 2 gives the running times in seconds to compute the distributions of all events of a complete day. Two trends are obvious: the wider the travel time distributions (increasing fluctuation
<table>
<thead>
<tr>
<th>parameter</th>
<th>$k = 1$</th>
<th>$k = 2$</th>
<th>$k = 3$</th>
<th>$k = 4$</th>
<th>$k = 5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>never</td>
<td>3.72s</td>
<td>6.23s</td>
<td>6.07s</td>
<td>6.98s</td>
<td>11.32s</td>
</tr>
<tr>
<td>rule-based</td>
<td>3.73s</td>
<td>6.30s</td>
<td>6.93s</td>
<td>7.02s</td>
<td>11.52s</td>
</tr>
<tr>
<td>static</td>
<td>4.12s</td>
<td>6.32s</td>
<td>7.01s</td>
<td>8.95s</td>
<td>11.94s</td>
</tr>
<tr>
<td>always</td>
<td>5.12s</td>
<td>7.29s</td>
<td>8.57s</td>
<td>10.99s</td>
<td>13.78s</td>
</tr>
</tbody>
</table>

Table 2: Running times in seconds for the different waiting strategies and travel time distributions.

parameter $k$), the larger the running time. Likewise, when we compare the rules among each other, we observe that the more trains have to wait for each other, the larger the running time. However, in all cases the absolute running times are below 14 s.

**Experiment 1.** This experiment investigates how well our predictions fit to realized data on two test days. It does not use any information about actual delays.

For this experiment we use the rule-based waiting rules. In Figure 4 one can observe the assessment of our forecasts for Thursday 10.03.2011 (left column). We measured the absolute differences between the expectation values and the realized times for 289459 available events. In the same manner we computed these data for Sunday, 20.02.2011 (with 193461 available events).

Note that we were only able to compare the available realized event data with our expectation values, see column 2 in Table 1. Obviously, we have more differences on Thursday. Our forecast seems to be not bad for further interpretations we consider another measure.

As a second measure for this scenario we count the number of realized timestamps which hit their corresponding $\kappa$-quantile. Figure 5 gives an impression of the assessment of our forecasts with respect to the daily situation.

The quantiles look quite similar for the two days. For smaller values of $\kappa$ the Thursday (10.3.2011) yields slightly better forecasts than the Sunday (20.02.2011). An explanation for this observation might be that the supports of the distribution on the Thursday are wider than on Sunday, thus it is easier to hit the support. With respect to expectation values, the difference to the realized values is less than 5 minutes in about 66% of all available events on both investigated days. However, a significant number of forecasts is wide off (by 2 hours in some cases) and the quantiles reveal a similar picture.

In order to interpret the results of this experiment, recall that our computation of arrival and departure time distributions is based on the pure schedule only, it does not incorporate actual delays. The comparison of our computed distributions and the actual delays clearly shows that our model is not capable to predict large source delays (for example, that a trains is delayed by two hours because of a defect of the engine).

**Experiment 2.** In this experiment we investigate the widths of all supports with respect to the time horizon. We compare again two different days, namely the 10.3.2011 and the 20.02.2011 in applying the above mentioned four different waiting rules. Figure 6 handles the waiting rule: always, Figure 7 the waiting rule: rule-based, Figure 8 the waiting rule: never and Figure 9 the waiting rule: static. The fluctuation parameter is again chosen as $k \in \{1, 2, 3, 4, 5\}$.

In Figure 6, we observe that an increasing widths of supports with an increasing time horizon if we use waiting rule always. Moreover, the larger the fluctuation parameter $k$ is, the larger are the influences of the increase of the widths, at an actual time and in future. If we compare the pictures on the left and on the right, we see considerable different expansions. The reason is that these days are different weekdays, namely a Thursday (10.03.2011) and a Sunday (20.02.2011). Obviously, the Thursday is more affected by disruptions as a relatively “silent” Sunday.

For the waiting strategy “rule-based” (see Figure 7), we observe similar effects as in the scenario always. Widths of all supports depend on the time horizon, fluctuation parameter $k$ and again on the type of weekday. However, the timetable is influenced much less in this realistic scenario. It seems to be relatively “robust” with respect to small fluctuations.
The results for the waiting rule “never” (see Figure 8) are pretty similar to those for the strategy “rule-based”. This can be explained by the fact that the standard waiting rules have a “no-wait” rule for many pairs of train categories.

For the waiting rule “static”, we observe that the width of the supports does not increase over time. Our goal was to study the precision of our forecasts, measured by the widths of the supports and with respect to the time horizon. In summary, with the exception of the waiting rule always which is obviously not well suited, all other rules behave in a reasonable way. Further analyses with actual data on realized delays are required to enable a finer differentiation among these rules.

5 Conclusions

We have presented a stochastic model for delay propagation in large transportation networks. This model turns out to be fast enough for an online scenario with massive streams of update messages.

In our initial experiments we worked with simple artificial distributions for travel time fluctuations (in the absence of real distributions). The next step is to replace these distributions by empirical distributions from collected statistical data over several months.

Furthermore, we will integrate online data on source delays. We expect that this will lead to much better delay forecasts. A more fine-grained analysis has to investigate how fast the predicting power decreases with the time horizon.

Further reasons for the limited forecasting power are due to several simplifications in our model, in particular, the fact that our model neglects the existence of capacity conflicts on tracks and uses a stochastic independency assumption.

Improved stochastic delay predictions has applications in the robustness analysis of timetables, timetable information systems and train disposition.

Acknowledgements. This work was partially supported by grants MU 1482/4-2 within the DFG Priority Programme SPP 1307 Algorithm Engineering. The authors wish to thank Deutsche Bahn AG for providing us with test data for scientific use.

References


Figure 4: Comparison of the absolute differences from the expected and realized timestamps on 10.03.2011 (left column) and on 20.02.2011 (right column) with increasing fluctuation parameter $k$ for travel time distributions.
Figure 5: Fraction of realized timestamps on Thursday 10.3.2011 (left column) and on Sunday 20.02.2011 (right column) in $\kappa$-quantile with varying $\kappa$ with increasing fluctuation parameter $k$ for travel time distributions.
Figure 6: Number of all distributions on 10.03.2011 (left) and on 20.02.2011 (right) with given constant widths of supports with respect to the time horizon and waiting rule: always.
Figure 7: Number of all distributions on 10.03.2011 (left) and on 20.02.2011 (right) with given constant widths of supports with respect to the time horizon and waiting rule: rule-based.
Figure 8: Number of all distributions on 10.03.2011 (left) and on 20.02.2011 (right) with given constant widths of supports with respect to the time horizon and waiting rule: never.
Figure 9: Number of all distributions on 10.03.2011 (left) and 20.02.2011 (right) with given constant widths of supports with respect to the time horizon and waiting rule: static.
Information about the Author(s)

Annabell Berger  
Institut für Informatik  
Martin-Luther-Universität Halle-Wittenberg  
06099 Halle, Germany  
E-Mail: berger@informatik.uni-halle.de

Andreas Gebhardt  
Institut für Informatik  
Martin-Luther-Universität Halle-Wittenberg  
06099 Halle, Germany  
E-Mail: gebhardt@informatik.uni-halle.de

Matthias Müller-Hannemann  
Institut für Informatik  
Martin-Luther-Universität Halle  
06099 Halle, Germany  
E-Mail: muellerh@informatik.uni-halle.de  
WWW: http://www.informatik.uni-halle.de/muellerh

Martin Ostrowski  
Institut für Informatik  
Martin-Luther-Universität Halle-Wittenberg  
06099 Halle, Germany  
E-Mail: martin.ostrowski@student.uni-halle.de