Modeling Time Table based Tram Traffic

Daniel Lückerath[1,2,*], Oliver Ullrich[1], Ewald Speckenmeyer[1]

[1]University of Cologne, Inst. f. Informatics, Pohligstrasse 1, 50969 Cologne, Germany;
[2]Cologne University of Applied Science, Colohne, Germany; **lueckerath@informatik.uni-koeln.de*

Abstract. In mid-sized cities, tram networks are major components of public service infrastructure. In those networks with their typically dense schedules, multiple lines share tracks and stations, resulting in a dynamic system behavior and mounting delays following even small disruptions. Robustness is an important factor to keep delays from spreading through the network and to minimize average delays.

This paper describes part of a project on simulation and optimization of tram schedules, namely the development and application of a simulation model representing a tram network and its assigned time table. We begin by describing the components of a tram network, which consist of physical and logical entities. These concepts are then integrated into a model of time table based tram traffic. We apply the resulting simulation software to our hometown Cologne's tram network and present some experimental results.

Introduction

Tram networks are important parts of public transport infrastructure, which is exemplified by the 745,000 passengers that are transported in Cologne's tram network every day as described in [5]. Especially mid-sized cities often have mixed tram networks, i.e. networks where trams travel on street level (thus being subject to individual traffic and corresponding traffic regulation strategies) and on underground tracks. In such dense networks robustness is an important factor to minimize average delays. Robustness measures the degree on which inevitable small disturbances, e.g. obstructed tracks due to parked cars, have impact on the whole network. With robust time tables delays are kept at a local level, whereas with non-robust time tables they spread through the network and might subsequently cause delays of other vehicles as described in [2, 3].

In this paper we present the simulation module (first described in [4]) which is part of a larger project to generate and evaluate robust time tables in order to minimize the impact of small delays. We develop a model and implement an application to simulate time tables of mixed tram networks in order to evaluate given time tables before their implementation in the field and to compare time tables generated by optimization methods (as in [7, 8]) with respect to their applicability. In addition we want to show that the adopted simulation engine can be applied to real world problems.

A more detailed description of our project and in particular our optimization approach is presented in the accompanying paper ‘Simulation and optimization of Cologne's tram schedule’ (see [7]).

We begin the remainder of this paper by describing the basics of time table based tram traffic (Section 1), followed by a short discussion of our model representing the physical and logical entities of the tram network (Section 2). The resulting software is then applied to Cologne's tram network (Section 3). We close with a short summary of the lessons learned and give some remarks on further research (Section 4).

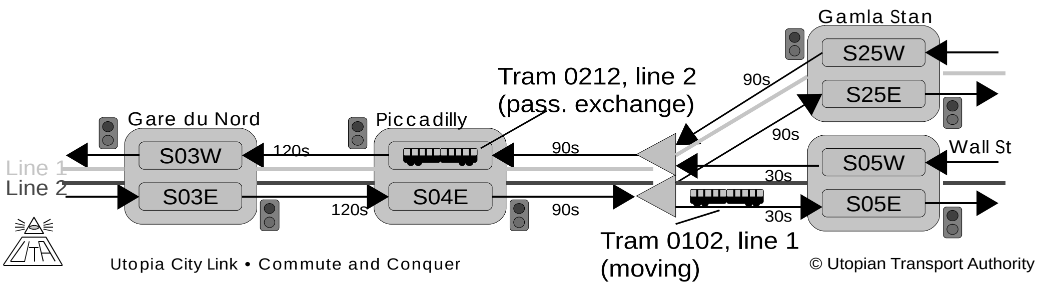

Figure 1. Part of a tram network.

SNE Simulation Notes Europe – Print ISSN 2305-9974 | Online ISSN 2306-0271
SNE 22(2), 2012, 61-68 | doi: 10.11128/sne.22.tn.10121

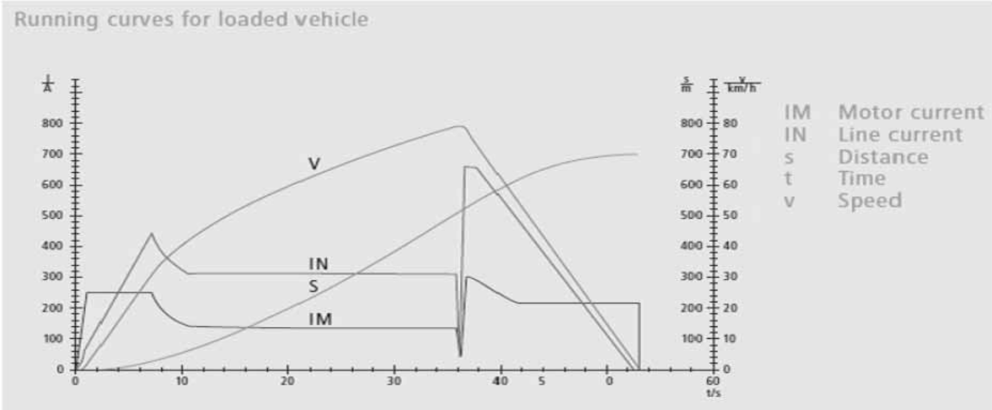

Figure 2. Maneuvering capabilities of wagon type K4000 as found in [10].

1 Time Table - based Tram Traffic

Tram networks can be considered as a combination of physical and logical components. The physical network consists of tangible entities, e.g. stations, tracks or trams, whereas the logical network is comprised of concepts and plans, e.g. lines, trips or time tables. Figure 1 shows an extract of an example network.

At the beginning of each *turn*, which is the planned movement of a vehicle through the network on a specific operational day, a tram leaves the *maintenance and storage depot* where it was stored over night. It then travels to the first *platform* of its first *trip*, where the passenger exchange takes place. Platforms are usually unidirectional and always part of a *station*, which combines adjacent platforms under a common name.

After executing the passenger exchange the vehicle travels to the next platform of the trip. The order of platforms which have to be visited is defined by the *line route*. Different line routes can be combined under a common name, thus constituting a *line*. For example Cologne's line 1 (from Junkersdorf to Bensberg and back) actually consists of 27 line routes, 15 of which are east bound and 12 are west bound.

The wagons used by the tram define the maneuvering capabilities and hence how it moves through the network. Table 1 depicts some important characteristics for the three different wagon types which are in use in Cologne's tram network and Figure 2 shows the maneuvering capabilities of wagon type K4000.

The *tracks* between two locations of the network are usually unidirectional, but bidirectional tracks also exist. Some tracks may have speed limitations due to their environment, e.g. inner-city tracks may have a speed limit because of traffic regulations.

While the vehicle travels from one platform to another it may have to traverse *track switches.* These are locations where more than two tracks meet; they can be differentiated between dividing and joining track switches. Like platforms and tracks, track switches are usually unidirectional. All but one of the tracks sharing one side of the track switch must form a curve, which leads to speed limitations that are usually lower than the speed limits on tracks.

The access to track switches (as well as to platforms and track sections) is usually controlled by *traffic lights.* At the end of the operational day the tram travels once again to a maintenance and storage depot. The spatial and chronological order of the vehicles in use on a specific operational day is constituted by the *time table*, i.e. the time table assigns each tram a turn and each turn a set of line routes with starting times.

Characteristics	**K4000**	**K4500**	**K5000**
Length of wagon	29.2 m	29.0 m	29.3 m
Weight of wagon	35.0 t	39.0 t	37.8 t
Maximum velocity	80 kph	80 kph	80 kph
Acceleration	1.3 m/s²	-	1.2 m/s²
Normal brake ability	1.4 m/s²	-	1.2 m/s²
Brake ability for emergency brake	3.0 m/s²	-	2.73 m/s²

Table 1. Characteristics of different wagon types as found in [10], [11] and [12].

2 Modeling Tram Traffic

2.1 Approach

Our approach to model and implement the described system is based on the characteristics of the adopted dynamic-adaptive parallel simulation engine (first described in [6]), which is still under development and was up to now tested on randomized graphs only.

The framework follows a model-based parallelization approach, which tries to exploit the embedded model's intrinsic parallelism. To take maximum advantage of this, the engine is limited to systems that can be considered as sparse, directed graphs, which include many traffic simulation models.

While building the model a number of the applied simulation engine's requirements have to be met. Each model node belongs at every instant to exactly one computational node, which can be a processor or processor core sharing a common cache with its neighbors, or a remote computer connected via a network by message passing. Communication takes place exclusively between computational nodes whose model nodes are connected via edges. The means of communication are transparent to the model nodes. Furthermore the simulation engine takes care of dynamic load balancing, its mechanics are beyond the scope of this paper and are described in [6].

2.2 Physical Network

The tram network is modeled as a directed graph with platforms, tracks and track switches represented by nodes. Every node administrates its currently hosted vehicles. Connections between nodes are represented as edges. Figure 3 depicts an example graph.

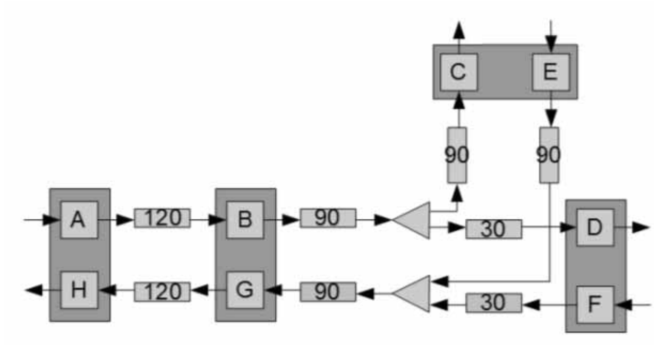

Figure 1. Example graph representing part of a tram network. Squares represent platforms, rectangles tracks and triangles track switches. The darker rectangles around platforms indicates that these platforms form a station.

At any point of time only one vehicle can be located at a platform, which is the main element for modeling boarding and deboarding of passengers. In the real world system passenger exchange is influenced by the platform and day time as well as tram type and passengers (e.g. speed and number). For simplicity's sake we model the boarding and deboarding of passengers as loading time distributions specific to platform and tram type with the combined duration of opening and closing the vehicle doors as minimum value.

Tracks are the only type of node that allows for more than one tram to be located at it at any point of time. The only exceptions to this rule are bidirectional tracks, which have to be exclusively reserved before a vehicle is allowed to enter them. Because the applied ÖPNV data model described in [9] does not allow for bidirectional connections between two locations of the network, they are modeled as two opposed unidirectional tracks. Reservation of one of the coupled tracks then causes blocking of the corresponding opposing track. Tracks also administrate traffic lights located on them.

As in [2] track switches are modeled as transfer points, i.e. they pass trams from an incoming to an outgoing track. Like platforms track switches can only be occupied by one tram at any point of time. Hence they have to be reserved before being entered and unblocked afterwards. Track switches are the only node type that can have more than two neighbors.

As described above, traffic lights are administrated by tracks. Their position at the track is given as an offset related to the beginning of the corresponding track. Phase change is modeled as a function. This is possible because in the described model each traffic light i has constant specific phase lengths t_{red} and t_{green} and subsequently equal cycle lengths

$$t_{cycle} = t_{red} + t_{green}$$

Randomly choosing the time of the first phase change from green to red by

$$t_{init}(i) = -1 * rand(0, t_{cycle}),$$
$$rand(0, t_{cycle}) \sim U(0, t_{cycle} - 1),$$

the current status can be calculated as given in following formula (1):

$$a: \mathbb{N} \rightarrow \{red, green\}$$

$$a_i(t) = \begin{cases} red, if\left(t - t_{init}(i)\right) \bmod t_{cycle} \leq t_{red} \\ green, else \end{cases} \quad (1)$$

Trams must always be located at a node of the network and their main attributes are specified by the type of wagons used. The tram type also holds functions for the maneuvering capabilities.

As an example the velocity during acceleration from zero as a function of time for tram type K4000 is shown in formula (2).

$$v(t) = \begin{cases} 0 & if\ t < 1 \\ \frac{14}{3} * t - \frac{10}{3} & if\ 1 \leq t < 8 \\ 35.33 * \sqrt[3]{t} - 36.66 & if\ 8 \leq t \leq 36 \\ 80 & else \end{cases} \quad (2)$$

Additional tram types can easily be included in the model by extending the abstract base class.

The tram submodel is based upon the event based simulation approach. Thus trams change their state at events of certain types, like stopping, or accelerating, which happen at discrete points of time. As a result of the event handling the system state may change and follow-up events are generated. Those are usually administrated in a priority queue, also called *Future Event List (FEL)*, as described in [1].

During the modeling process fourteen event types were identified (see Table 2).

Trip start	Emergency brake start
Trip end	Acceleration start
Tram standing	Passenger exchange start
Movement start	Track switch reservation
Braking start	Free track switch
Crash	Bidirectional track reservation
Transfer to next node	Free bidirectional track

Table 2. Identified types of simulation events.

As an example Listing 1 shows the handling of event "tram standing" in pseudo code.

2.3 Logical Network

Most parts of the logical network do not have to be modeled explicitly, i.e. a line just combines a set of line routes under a common name and hence can be implemented as a simple string or integer value.

```
Event "tram standing" for tram t do
 if t is located at a stop then
   if passenger exchange completed then
     try to transfer t to next node
         (and if necessary allocate
          following bidirectional
track)
     catch failed transfer by remaining
           to wait for n seconds
   else execute passenger exchange
 else if t is located on a track then
   if t has reached end of track then
     try to transfer t to next node
         (and if necessary allocate
          following switch)
     catch failed transfer by remaining
           to wait for n seconds
   else accelerate
```

Listing 1. Pseudo code algorithm for event type "tram standing".

A line route on the other hand holds more information and therefore is modeled explicitly. Main component of a line route is a sorted list of identifiers of platforms which have to be visited in this order. Because the ÖPNV data model contains no information about track switch locations on line routes, this information has to be computed prior to the simulation or dynamically before a tram tries to transfer to the next node. In order to identify individual line routes, each one is assigned a name and a unique ID.

Trips allocate a planned starting time to a specific line route and are assigned unique IDs. Each tram then holds a sorted list of trips, which constitutes its turn. The set of turns of a specific operational day constitutes the time table of that day.

2.4 Simulation Infrastructure

In order to meet the requirements of the parallel simulation engine the tram network is divided into disjoint parts, each of which is then allocated to a model node. The special case of assigning the whole network to one model node results in a sequential simulation.

Each model node holds a priority queue of trams located on the part of the network allocated to the node.

When the model node receives the instruction to calculate the next simulation step it first inserts new vehicles, i.e. trams that were sent by neighboring model nodes, into the priority queue. It then instructs each vehicle whose time stamp is equal to the simulation time to execute the next simulation step.

Finally all vehicles that need to be transferred are sent to neighboring model nodes.

3 Simulating Cologne's Tram Network

We apply the developed simulation software to our hometown Cologne's tram network based on the time table data of 2001, as seen in Figure 4. It consists of 528 platforms and 58 track switches connected via 584 tracks. These tracks cover a total length of 407.4 kilometers, resulting in an average track length of 697.6 meters. 15 lines with 182 line routes exist. On each operational day 2,814 trips are executed by 178 trams.

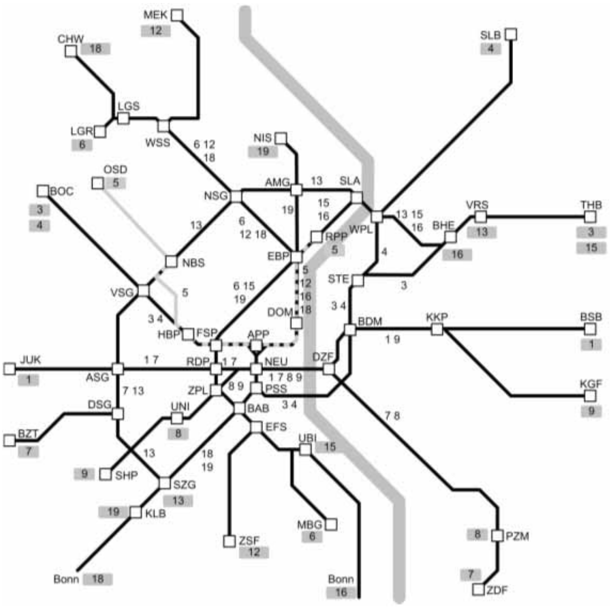

Figure 4. Cologne's tram network in 2001.

We map each node of the graph representing the tram network as a model node and execute 100 simulation runs, yielding an average run time of 348 seconds for a whole operational day.

The results show an average delay of departure over the whole system of 18.67 seconds and a mean delay of 36.05 seconds. During the whole operational day 39,674 departure delays occur, of which 32,389 (81.6%) are less than or equal to 60 seconds (see Figure 5).

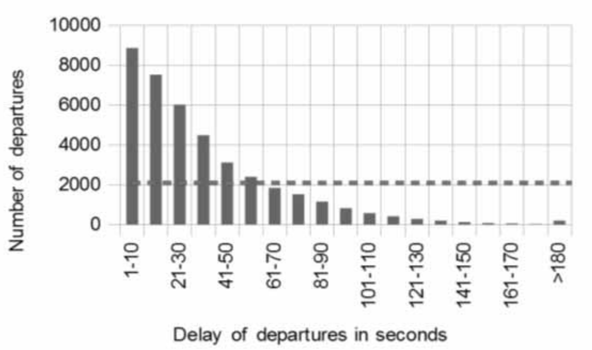

Figure 5. Delay frequencies.

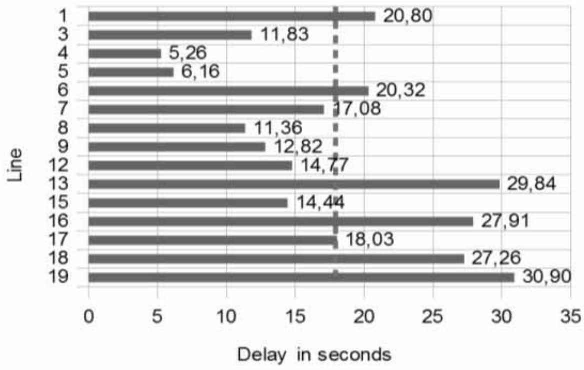

Figure 6. Line delay.

As seen in Figure 6 the lines of the network vary greatly in average delay, mainly due to differences in route length, departure frequencies and inter line dependencies.

For the remainder of this paper we take a closer look at line 5 (see highlighted line in Figure 4) in order to confirm plausibility of our model and to show that the results of our application reflect phenomena observable in Cologne's tram network. Serving 17 platforms line 5 is the shortest line of the network and therefore best qualified for a detailed discussion. About half of the line runs through the inner city, while the other half runs through suburbs. It shares most of its inner city tracks with lines 3, 4, 12, 16, 18 and short parts also with lines 6, 13, 15 and 19. Furthermore for about one third of its tracks line 5 travels underground.

Figure 7 depicts the average delay over the served platforms of trip no. 6 of tram 504, starting at 7:47 at Ossendorf station (OSD) and traveling to Reichenspergerplatz (RPP). During the first half of its trip the tram travels along tracks not shared with other lines.

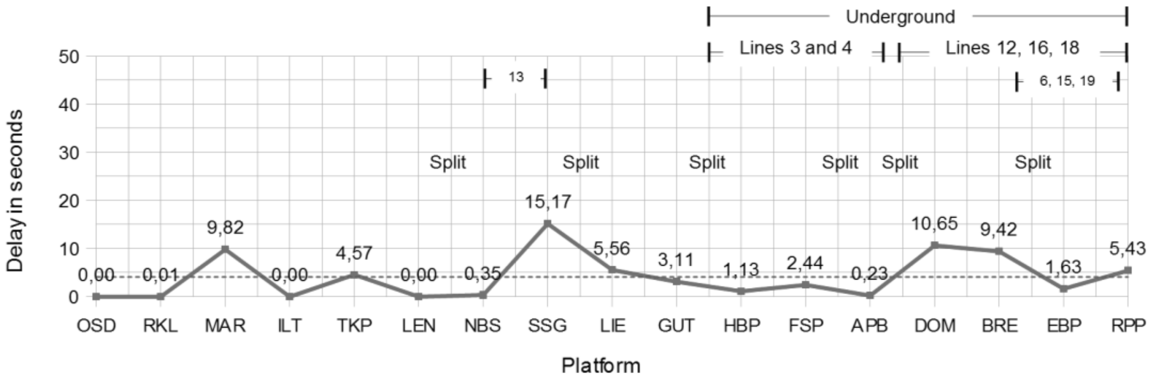

Figure 7. Line 5, Tram 504, Trip 6, starting at 7:47 at Ossendorf.

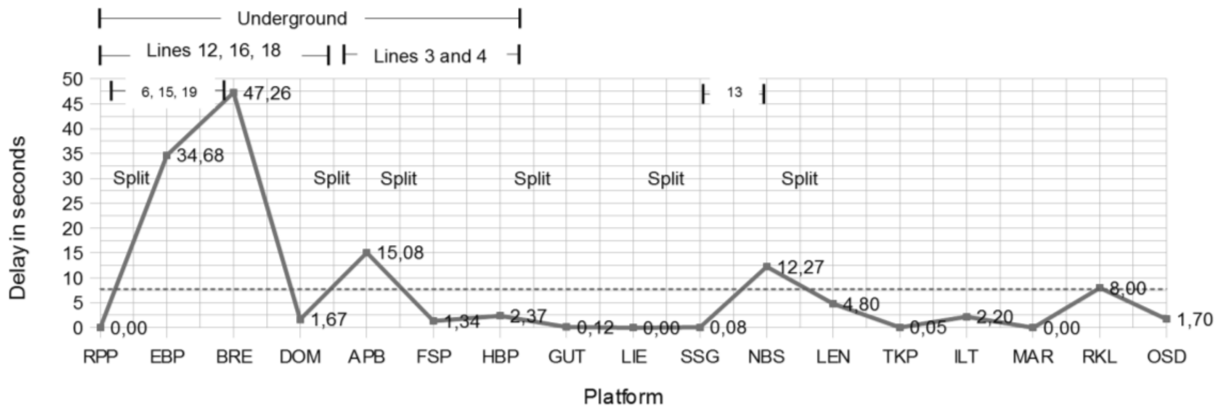

Figure 8. Line 5, Tram 504, Trip 7, starting at 8:21 at Reichenspergerplatz.

The first two peaks in delay at stations Margaretastrasse (MAR) and Takuplatz (TKP) result from a too tight schedule, i.e. the tram needing more than the scheduled 60 seconds to traverse the 700 meter and 580 meter tracks leading to MAR and TKP. On the other hand the planned travel times to the succeeding stations are twice as high, while both tracks are roughly 100 meter shorter. Thus the vehicle is able to eliminate the delay completely.

Though with a length of 280 meter shorter than e.g. the track leading to MAR and having the same planned travel time (60 seconds), a similar effect can be observed between stations Nussbaumerstrasse (NBS) and Subbelratherstrasse/Gürtel (SSG). This is due to the fact that the tram has to pass two traffic lights on the way.

Because traffic lights in the described model have constant phase lengths, the average waiting time t_w at each traffic light can be calculated as seen in following formula (3):

$$t_w = \frac{t_{green}}{t_{cycle}} * 0 + \frac{t_{red}}{t_{cycle}} * \frac{t_{red}}{2} \qquad (3)$$

For our experiments we assumed $t_{red} = t_{green} = 30$ seconds, hence from NBS to SSG the tram has to wait 2 * 7.5 = 15 seconds on average, leaving only 45 seconds to traverse the track, coordinate with joining line 13 and exchange pasengers at SSG.

Between SSG and Hans-Böckler-Platz (HBP) the vehicle is able to reduce the delay. The reduction rate flattens after station Liebigstrasse (LIE) because the tram has to pass traffic lights once again. Furthermore after Gutenbergstrasse (GUT) the tram has to coordinate with vehicles of joining lines 3 and 4.

After station Appellhofplatz (APB) lines 3 and 4 separate from line 5 and lines 12, 16 and 18 join. The necessary coordination between the vehicles results in the accumulation of delay at station Dom/Hbf (DOM).

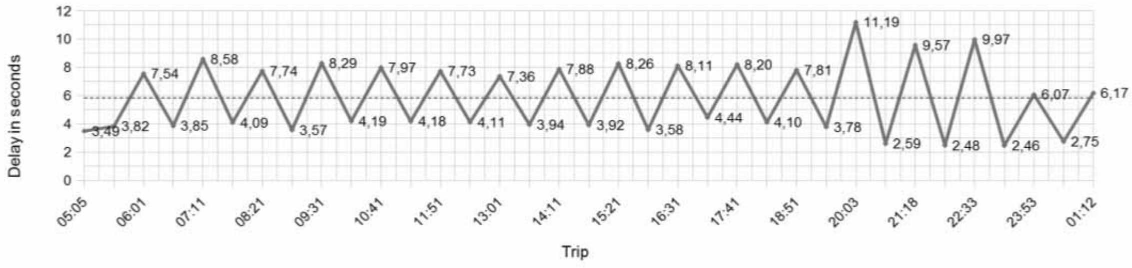

Figure 9. Average delay of trips of tram 504.

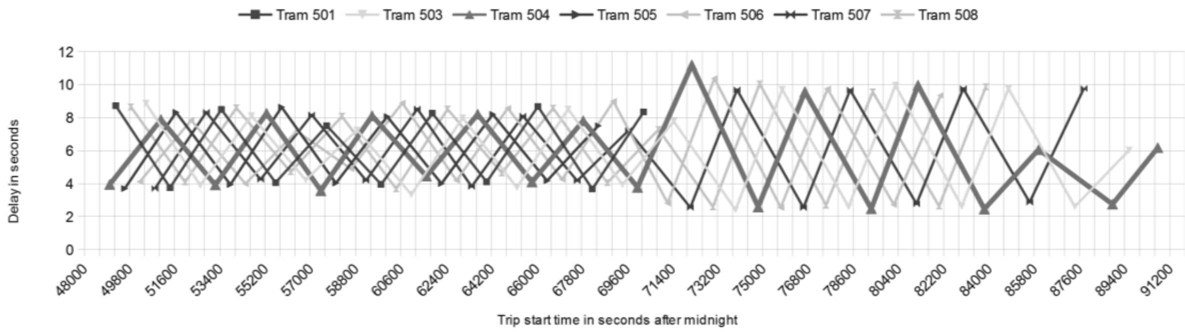

Figure 10. Delay of all trips of line 5 since 13:20.

Figure 8 shows the follow-up trip of tram 504. The increase in delay between RPP and Ebertplatz (EBP) in contrast to the more moderate during the preceding trip can be explained by the significantly smaller safety distance between lines 5 and 18 (one minute compared to three minutes). From Breslauer Platz (BRE) to DOM the vehicle is able to reduce its delay almost completely, while in the opposite direction no such effect can be observed. The cause of this is that the planned travel time from BRE to DOM is 60 seconds higher than the travel time for the opposite direction, accounting for a higher expected time for passenger exchange at Dom/Hbf, which is a major national railway node. Because our model currently does not account for this the simulated vehicle is able to reduce the delay.

Since no vehicle leaves its current platform ahead of the planned departure time no travel time buffer is aggregated, as can be seen between GUT and LIE, where the delay could not be reduced below zero

Observing a vehicle over a whole operational day (tram 504, Figure 9) we see a clear pattern: every trip from RPP to OSD has a higher average delay than trips from OSD to RPP. The only exception to this is the first trip of the operational day which is a short maintenance trip.

The average delay of trips from RPP to OSD is higher than the average delay of trips in the opposite direction, because vehicles traveling from RPP to OSD accumulate a very high delay over the first three platforms where the coordination between lines 5, 6, 12, 15, 16, 18 and 19 is amiss. On the other hand, during trips from OSD to RPP the coordination between vehicles at the critical platforms is considerably better, resulting in a lower average delay.

During the evening hours of the operational day, beginning at 20:00 o'clock, a change in the delay amplitude can be observed (see Figures 9 and 10). The cause of this is twofold. First the tact of the schedule is changed from 10 to 15 minutes in order to reflect lesser demand. Secondly, as a result of the change in tact vehicles are taken out of the system. Thus trams of all lines head for the maintenance and storage depots, which are located at central points in the network, resulting in an increase in utilization of tracks leading to those depots. This worsens the already poor coordination between lines on the outbound tracks. After the second tact change (from 15 to 30 minutes) at roughly 23:00 o'clock coordination between the remaining vehicles gets better again.

Both conditions can be observed for all trams as can be seen in Figure 10, which depicts the average trip delay for all vehicles of line 5 between 13:20 and 01:10.

4 Conclusions and Future Work

In this paper we described our approach for modeling time table based tram traffic. Beginning with a description of the structure of tram networks, which can be considered as a combination of physical and logical components, we described the different entities, e.g. trams, tracks or traffic lights, and their interaction.

After that we characterized our approach for modeling tram networks as graphs with trams as transient entities encapsulating most of the event based simulation logic, using the parallelization framework.

Finally we applied the developed simulation software to Cologne's tram network and analyzed some results. We were able to demonstrate that our application shows the expected behavior and the results reflect the phenomena observable in Cologne's tram network. We also demonstrated real world applicability of the simulation engine.

In further steps the developed model will be applied to other time tables generated with the help of optimization tools as well as real world time tables for further evaluation. First results can be found in the accompanying paper “Simulation and optimization of Cologne's tram schedule” (again, see [7]).

Acknowledgements

This work was partially supported by *Rhein Energie Stiftung Jugend Beruf Wissenschaft* under grant number W-10-2-002. We thank the other members of our work group Patrick Kuckertz and Bert Randerath.

An abbreviated version of this paper was previously published in Bödi, R., Maurer W. (Ed.): Proceedings of ASIM 2011, Winterthur, 2011.

References

[1] J. Banks, J.S. Carson, B.L. Nelson, D.M. Nicol. *Discrete-Event System Simulation*. Upper Saddle River, Pearson, 2010.

[2] G. Lückemeyer. *A Traffic Simulation System Increasing the Efficiency of Schedule Design for Public Transport Systems Based on Scarce Data*. Shaker Verlag, Aachen, 2007.

[3] G. Lückemeyer, E. Speckenmeyer. *Comparing Applicability of Two Simulation Models in Public Transport Simulation.* In M. Becker, H. Szczerbicka (Ed.): ASIM 2006 – 19. Symposium Simulationstechnik. ASIM/Universität Hannover, 2006.

[4] D. Lückerath. *ParSiVaL – Entwurf und Entwicklung einer Anwendung zur parallelen Simulation von schienengebundenem Öffentlichen Personennahverkehr*. Diplomarbeit, Universität zu Köln, 2011.

[5] Oberbürgermeister der Stadt Köln. *Stadtentwicklung in Köln – Mobilitätsentwicklung in Köln bis 2025*. Stadt Köln, 2008.

[6] O. Ullrich. *Eine dynamisch-adaptive Ansatz zur modellbasierten Parallelisierung von Simulationsanwendungen.* Work in progress, Universität zu Köln.

[7] O. Ullrich, D. Lückerath, E. Speckenmeyer. *Simulation and Optimization of Cologne's tram schedule*. SNE Simulation Notes Europe, 22(2), 67-74, ARGESIM Vienna 2012. doi: 10.11128/sne.22.tn.10123

[8] E. Speckenmeyer, N. Li, D. Lückerath, O. Ullrich. *Socio-economic Objectives in Metro Scheduling* (Technical Report). Zentrum für angewandte Informatik Köln, 2011.

[9] Verband Deutscher Verkehrsunternehmen eV. *VDV-Standardschnittstelle Liniennetz/Fahrplan.* VDV-Schriften 452 (2008).

[10] Vossloh Kiepe GmbH. *Elektrische Ausrüstung des Niederflur-Stadtbahnwagens K4000 der Kölner Verkehrs-Betriebe AG*. Druckschrift 00KV7DE, 2003.

[11] Vossloh Kiepe GmbH. *Elektrische Ausrüstung des Niederflur-Stadtbahnwagens K4500 für die Kölner Verkehrs-Betriebe AG.* http://www.vossloh-kiepe.com/vkproduktordner.2008-05-14.1154367607/vkproduktordner.2008-06-30.8585393121/vkproduktordner.2008-05-15.5609169940/vkprodukt.2008-05-14.8640403215/vkprodukt_download accessed on 24.05.2011.

[12] Vossloh Kiepe GmbH. *Elektrische Ausrüstung der Hochflur-Stadtbahnwagen K5000 der Kölner Verkehrs-Betriebe AG.* http://www.vossloh-kiepe.com/vkproduktordner.2008-05-14.1154367607/vkproduktordner.2008-06-30.8585393121/vkproduktordner.2008-05-15.5609169940/vkprodukt.2008-06-04.1250026636/vkprodukt_download accessed on 24.05.2011.

Submitted: September 2011
Revised: July, 2012
Accepted: August 1, 2012