Transport and Telecommunication, 2009, Volume 10, No 2, 33–42
Transport and Telecommunication Institute, Lomonosova 1, Riga, LV-1019, Latvia

RELIABILITY AND FUNCTIONAL ANALYSIS OF DISCRETE TRANSPORT SYSTEMS BY MODELLING AND SIMULATION

Tomasz Walkowiak, Jacek Mazurkiewicz

Institute of Computer Engineering, Control and Robotics
Wroclaw University of Technology
ul. Janiszewskiego 11/17, 50-372 Wroclaw, Poland
E-mails: Tomasz.Walkowiak@pwr.wroc.pl, Jacek.Mazurkiewicz@pwr.wroc.pl

The paper describes a novel approach to the analysis of discrete transport systems (DTS). The proposed method is based on modelling and simulating of the system behaviour. Monte Carlo simulation is a tool for DTS performance metric calculation. No restriction on the system structure and on a kind of distribution is the main advantage of the method. The system is described by the formal model, which includes reliability and functional parameters of DTS. The paper presents a description of developed simulator with its performance analysis. Moreover a case study based on real transport system with exemplar results is given.

Keywords: reliability, discrete transport system, Monte-Carlo simulation

1. Introduction

Administration of a large transport system is not a trivial task. The transport systems are characterized by a very complex structure. The performance of the system can be impaired by various types of faults related to the transport vehicles, communication infrastructure or even by traffic congestion [1]. It is hard for human (administrator, owner) to understand the behaviour of the system. To overcome this problem we propose a functional approach. The transport system is analysed from the functional point of view, focusing on business service realized by a system [2]. The analysis is following a classical [3]: modelling and simulation approach. It allows calculating different system measures, which could be a base for decisions related to administration of the transport systems. The metric are calculated using Monte Carlo techniques [4]. No restriction on the system structure and on a kind of distribution is the main advantage of the method. The proposed model allows forgetting about the classical reliability analysis based on Markov or Semi-Markov processes [5] – idealized and hard for reconciliation with practice.

The paper presents an analysis of transport system of the Polish Post regional centre of mail distribution (described in section 2). Base on which we have developed the discrete transport system model presented in section 3. The delivery of mails is the main service given by the post system. From the client's point of view the quality of the system could be measured by the time of transporting the mail from the source to destination. Therefore, the quality of the analysed system is measured by the availability defined as an ability to realize the transportation task at a required time (described in section 4).

The post system is very hard to be analysed by a formal model since it does not lie in the Markov process framework. Therefore, we have used a computer simulation [3] described in section 5. Next (section 6), we give an example of using presented model and simulator for the analysis of the Polish Post regional centre in Wroclaw transport system and discussed the performance of developed simulator.

2. Transport System of Polish Post

The analysed transport system is a simplified case of the Polish Post. The business service provided the Polish Post is the delivery of mails. The system consists of a set of nodes placed in different geographical locations. Two kinds of nodes could be distinguished: central nodes (CN) and ordinary nodes (ON). There are bidirectional routes between nodes. Mails are distributed among ordinary nodes by trucks, whereas between central nodes by trucks, railway or by plain. The mail distribution could be understood by tracing the delivery of some mail from point A to point B. At first the mail is transported to the nearest to A ordinary node. Different mails are collected in ordinary nodes, packed in larger units called containers and then transported by trucks scheduled according to some time-table to the nearest central node. In central node containers are repacked and delivered to appropriate (according to delivery address of each mail) central node. In the Polish Post there are 14 central nodes and more than 300 ordinary nodes. There are more than one million mails going through one central node within 24 hours. It gives a very large system to be modelled and simulated. Therefore, we have decided to model only a part of the Polish Post transport system – one central node with a set of ordinary nodes.

Essential in any system modelling and simulation is to define the level of details of modelled system. Increasing the details causes the simulation becoming useless due to the computational complexity and a large number of required parameter values to be given. On the other hand a high level of modelling could not allow recording required data for system measure calculation. Therefore, the crucial think in the definition of the system level details is to know what kind of measures will be calculated by the simulator. Since the business service given by the post system is the delivery of mails on time. Therefore, we have to calculate the time of transporting mails by the system. Since the number of mails presented in the modelled system is very large and all mails are transported in larger amounts containers, we have decided to use containers as the smallest observable element of the system. Therefore, the main observable value calculated by the simulator will be the time of container transporting from the source to the destination node.

The income of mails to the system, or rather containers of mails as it was discussed above, is modelled by a stochastic process. Each container has a source and destination address. The central node is the destination address for all containers generated in the ordinary nodes. Where containers addressed to any ordinary nodes are generated in the central node. The generation of containers is described by some random process. In case of central node, there are separate processes for each ordinary node. Whereas, for ordinary nodes there is one process, since commodities are transported from ordinary nodes to the central node or in the opposite direction.

The containers are transported by vehicles. Each vehicle has given capacity – maximum number of containers it can haul. Central node is a base place for all vehicles. They start from the central node and the central node is the destination of their travel. The vehicle hauling a commodity is always fully loaded or taking the last part of the commodity if it is less than its capacity. Vehicles operate according to the time-table. The time-table consists of a set of routes (sequence of nodes starting and ending in the central node, times of leaving each node in the route and the recommended size of a vehicle). The number of used vehicle and the capacity of vehicles do not depend on temporary situation described by number of transportation tasks or by the task amount for example. It means that it is possible to realize the route by completely empty vehicle or the vehicle cannot load the available amount of commodity (the vehicle is to small). Time-table is a fixed element of the system in observable time horizon, but it is possible to use different time-tables for different seasons or months of the year.

Summarizing the movement of the containers in the system, a container is generated with destination address in some of node (source) at some random time. Next, the container waits in the node for a vehicle to be transported to the destination node. Each day a given time-table is realized, it means that at a time given by the time-table a vehicle, selected from vehicles available in the central node, starts from central node and is loaded with containers addressed to each ordinary nodes included in a given route. This is done in a proportional way. When a vehicle approaches the ordinary node it is waiting in an input queue if there is any other vehicle being loaded/unload at the same time. There is only one handling point in each ordinary node. The time of loading/unloading vehicle is described by a random distribution. The containers addressed to given node are unloaded and empty space in the vehicle is filled by containers addressed to a central node. Next, the vehicle waits till the time of leaving the node (set in the time-table) is left and starts its journey to the next node. The operation is repeated in each node on the route and finally the vehicle is approaching the central node when it is fully unloaded and after it is available for the next route. The process of vehicle operation could be stopped at any moment due to a failure (described by a random process). After the failure, the vehicle waits for a maintenance crew (if there are no available due to repairing other vehicles), is being repaired (random time) and after it continues its journey. The vehicle hauling a commodity is always fully loaded or taking the last part of the commodity if it is less than its capacity.

3. Formal Model of the Discrete Transport System

3.1. Model overview

The described in the previous section regional part of the transport system of Polish Post with one central node and several ordinary nodes was a base for a definition of a formal model of a discrete transport system (DTS).

Generally speaking users of the transport system are generating tasks, which are being realized by the system. The task to be realized requires some services presented in the system. A realization of the system service needs a defined set of technical resources. Moreover, the operating of vehicles transporting mails between system nodes is done according to some rules – some management system. Therefore, we can model discrete transport system as a 4-tuple:

$$DTS = \langle Client, BS, TI, MS \rangle \tag{1}$$

$Client$ – client's model,
BS – business service, a finite set of service components,
TI – technical infrastructure,
MS – management system.

3.2. Technical infrastructure

During modelling of technical infrastructure we have to take into consideration functional and reliability aspects of the post transport system. Therefore, the technical infrastructure of DTS could be described by three elements:

$$TI = \langle No, V, MM \rangle, \tag{2}$$

where: No – set of nodes; V – set of vehicles; MM – maintenance model.

Set of nodes (No) consists of single central node (CN), a given number of ordinary nodes (ON_i). The distance between each two nodes is defined by the function:

$$distance : No \times No \rightarrow R_+ . \tag{3}$$

Each node has one functional parameter the mean (modelled by normal distribution) time of loading a vehicle:

$$loading : No \rightarrow R_+ . \tag{4}$$

Moreover, the central node (CN) has additional functional parameter: number of service points (in each ordinary node there is only one service point):

$$servicepoints : CN \rightarrow N_+ . \tag{5}$$

Each vehicle is described by the following functional and reliability parameters:

- mean speed of a journey

$$meanspeed : V \rightarrow R_+ , \tag{6}$$

- capacity – number of containers which can be loaded

$$capacity : V \rightarrow R_+ , \tag{7}$$

- mean time to failure

$$MTTF : V \rightarrow R_+ , \tag{8}$$

time when failure occurs is given by exponential distribution with mean equal to a value of $MTTF$ function,

- mean repair time

$$MRT : V \rightarrow R_+ . \tag{9}$$

The traffic is modelled by a random value of vehicle speed and therefore the time of vehicle (v) going from one node (n_1) to the other (n_2) is given by a formula:

$$time(v,n_1,n_2) = \frac{distance(n_1,n_2)}{Normal(meanspeed(v), 0.1 \cdot meanspeed(v))}, \tag{10}$$

where *Normal* denotes a random value with the Gaussian distribution.

Maintains model (*MM)* consists of a set of maintenance crews which are identical and unrecognised. The crews are not combined to any node, are not combined to any route, they operate in the whole system and are described only by the number of them. The time when a vehicle is repaired is equal to the time of waiting for a free maintains crew (if all crews involved into maintenance procedures) and the time of a vehicle repair which is a random value with the Gaussian distribution ($Normal(MRT(v), 0.1 \cdot MRT(v))$.

3.3. Business service

Business service (*BS*) is a set of services based on business logic that can be loaded and repeatedly used for concrete business handling process. Business service can be seen as a set of service components and tasks that are used to provide service in accordance with business logic for this process. Therefore, *BS* is modelled a set of business service components (*sc*),):

$$BS = \{sc_{1,}...,sc_n\}, n \ = length(BS) > 0, \tag{11}$$

the function *length*(*X*) denotes the size of any set or any sequence *X.*

Each service component in DTS consist of a task of delivering a container from a source node to the destination one.

3.4. Client's model

The service realised by the clients of the transport system are sending mails from a source node to a destination one. Client's model consists of a set of clients (*C*).

Each client is allocated in one of nodes of the transport system:

$$allocation: C \to No. \tag{12}$$

A client allocated in an ordinary node is generating containers (since, we have decided to monitor containers not separate mails during simulation) according to the Poisson process with destination address set to ordinary nodes. In the central node, there is a set of clients, one for each ordinary node. Each client generates containers by a separate Poisson process and is described by intensity of container generation:

$$intensity : C \to R_+. \tag{13}$$

The central node is the destination address for all containers generated in ordinary nodes.

3.5. Management system

The management system (*MS*) of the DTS controls the operation of vehicle. It consists of a sequence of routes:

$$MS = \langle r_1, r_2,..., r_{nr} \rangle. \tag{14}$$

Each route is a sequence of nodes starting and ending in the central node, times of leaving each node in the route (t_i) and the recommended size of a vehicle (*size*) :

$$r = \langle CN, t_0, n_1, t_1,..., n_m, t_m, CN, size \rangle \quad v_i \in No - \{CN\} \quad 0 \le t_0 < t_1 < ... < t_m < 24h$$

The routes are defined for one day (so, all times are values less than 24 h of leaving the node) and are repeated each day.

The management system selects vehicles to realise each route in random way, first of all vehicles (among vehicles available in central node) with capacity equal to recommended size are taken into consideration. If there is no such vehicle, vehicles with larger capacity are taken into consideration. If still there is no vehicle fulfilling requirements vehicle from vehicles with smaller size is randomly selected. If there is no available vehicle a given route is not realized.

4. Functional Metrics of DTS

4.1. Metrics overview

The formal model described above was designed to allow developing a simulator (described in the next section) which allows observing the time of transporting each container. Based on these observations several metrics could be defined.

As it was mentioned in the introduction we focus here on the service oriented approach [2]. Therefore we propose that the availability will be a key parameter for the evaluation of the quality of the DTS.

One can define the availability in different ways, but always the value of availability can be easy transformed into economic or functional parameters perfectly understood by owner of the system.

The availability is mostly understood as a probability that a system is up; and is defined as a ratio of the expected value of the uptime of a system to the observation time. It is a simple definition but requires defining what does it mean that transport system is working. The similar metric is the acceptance ratio defined in information since as a number of accepted requests to the total number of requests.

4.2. Functional availability

In paper [6] we have proposed the definition of up time as a time when the number of delayed containers does not exceed a given threshold. Let introduce the following notation:

- T – a time measured from the moment when the container was introduced to the system to the moment when the container was transferred to the destination (random value),
- T_g – a guaranteed time of delivery, if exceeded the container is delayed.
- $N_{deleyed}(t)$ – a stochastic process describing the number of delayed containers at time t, i.e. the number of containers for which T> T_g .

Therefore, the functional availability $A_k(t)$ can be defined as a probability that the number of delayed containers at time t does not exceed k, the value k is the level of acceptable delay:

$$FA_k(t) = \Pr\{N_{deleyed}(t) \le k\}. \tag{15}$$

4.3. Average functional availability

The defined in the previous section functional availability describes a state of an analysed system at a given point of time. In case if somebody wants to analyse a state of system in a time interval we have proposed in [7] the other metric: average functional availability $AFA_k(t)$. It is defined as an average probability that a system in the time interval from 0 to t is in up-time state (i.e. the number of delayed containers does not exceed threshold k):

$$AFA_k(t) = \frac{1}{t}\int_0^t \Pr\{N_{deleyed}(\tau) \le k\} d\tau . \tag{16}$$

4.4. Acceptance ratio

In [8] we have proposed other performance metric – acceptance ratio. It is defined as a ratio of on-time containers (containers for which T< T_g) to all containers within a given time period (24h was used). Therefore a sequence of time moments ($\tau_0, \tau_2, ..., \tau_i, ..., \tau_n$) when the metric is calculated has to be set – a midnight of each day was used (i.e. $\tau_i - \tau_{i-1} = 24h$) . Within each time period (τ_{i-1}, τ_i), a given number of containers are delivered ($N_{delivered}(\tau_{i-1}, \tau_i)$), a part of if or all delivered on time ($N_{ontime}(\tau_{i-1}, \tau_i)$), but at the end of analysed period time there could be some containers not yet delivered (waiting in the source node or being transported) $N_{insystem}(\tau_i)$ and all or part of them being not late yet ($N_{ontimeinsystem}(\tau_i)$). Taking into consideration introduced symbols the availability could be calculated as the expected value (Monte-Carlo approach) of ratio of on-time containers to all containers:

$$AR\ (t_i) = E\left(\frac{N_{ontime}(\tau_{i-1},\tau_i) + N_{ontimeinsystem}(\tau_i)}{N_{delivered}(\tau_{i-1},\tau_i) + N_{insystem}(\tau_i)}\right). \tag{17}$$

5. Discrete Transport System Simulation

5.1. Event-driven simulation

Discrete transport system described in the previous section is very hard to be analysed by formal methods. It does not lie in the Markov process framework [5]. A common way of analysing that kind of systems is a computer simulation. To analyse the system we must first build a simulation model, which was done based on the formal model presented in the previous section, and then operate the model. The system model needed for simulation has to encourage the system elements behaviour and interaction between elements.

Once a model has been developed, it is executed on a computer. It is done by a computer program, which steps through time. One way of doing it is the so-called event-driven simulation. It is based on an idea of event, which could be described by time of event occurring and type of an event. The simulation is done by analysing a queue of event (sorted by time of event occurring) while updating the states of system elements according to rules related to a proper type of event. Due to a presence of randomness in the DTS model the analysis of it has to be done based on Monte-Carlo approach [4]. It requires a large number of repeated simulations.

Summarizing, the event-driven simulator repeats *N*-times the following loop:

- beginning state of a DTS initialisation,
- event state initialisation, set time $t = 0$,
- repeat until $t < T$:
- take first event from event list,
- set time equals time of event,
- realize the event – change state of the DTS according to rules related to proper type of event: change objects attributes describing system state, generate new events and put them into event list, write data into output file.

5.2. Events and elements of DTS simulator

In case of DTS following events (mainly connected with vehicles) have been defined:

- vehicle failure,
- vehicle starts repair,
- vehicle repaired,
- vehicle reached the node,
- vehicle starts from the node,
- vehicle is ready for the next route,
- time-table (starting the route in the central node).

The processing of events done in objects is representing DTS elements. The objects are working in parallel. The following types of system elements are distinguished: vehicle, ordinary node, central node, time-table.

The life-cycle of each object consists of waiting for an event directed to this object and then execution of tasks required to perform the event. These tasks includes the changes of internal state of the object (for example when vehicle approaches the node it is unloaded, i.e. the number of hauled containers decreases) and sometimes creating a new even (for example the event vehicle starts from the node generates new event vehicle reached the node – next node in the trip). The random number generator is used to deal with random events, i.e. failures. It is worth to notice that the current analysed event not only generates a new event but also could change time of some future events (i.e. time of approaching the node is changed when failure happens before). The time of a new event is defined by the sum of current time (moment of execution of the current event) and the duration of a given task (for example, vehicle repair). Only times of starting a given route (event vehicle starts from the central node) are predefined (according to the time-table). Duration of all other tasks are defined by system elements states:

- time when vehicle waits in the queue for loading/unloading,
- time when vehicle waits in the queue for maintains crew,
- or are given by random processes:
- time of vehicle going between two nodes,
- time of loading/unloading,
- time to failure,
- repair time.

Moreover each object representing a node have additional process (working in parallel) which are responsible for generating containers. The life cycle of this process is very simple: waiting a random time, generating a container with a given destination address (central node for all ordinary nodes, and each ordinary nodes for process in the central node) and storing a container in the store house (implemented as a queue) of a given node.

5.3. DTS simulator implementation

The event-simulation program could be written in a general purpose programming language (like C++), in a fast prototyping environment (like Matlab) or a special purpose discrete-event simulation kernel. One of such kernels, is the Scalable Simulation Framework (SSF) [9] which is a used for SSFNet [9,10] computer network simulator. SSF is an object-oriented API - a collection of class interfaces with prototype implementations. It is available in C++ and Java. SSFAPI defines just five base classes: Entity, inChannel, outChannel, Process, and Event. The communication between entities and delivery of events is done by channels (channel mappings connects entities).

For the purpose of simulating DTS we have used Parallel Real-time Immersive Modelling Environment (PRIME) [10] implementation of SSF due to a much better documentation then available for the original SSF. We have developed a generic class derived from SSF Entity which is a base of classes modelling

DTS objects which models the behaviour of presented in section 2 and 3 discrete transport system.

As it was mentioned a presence of randomness in the DTS model, the Monte-Carlo approach is used. The original SSF was not designed for this purpose so some changes in SSF core were done to allow to restart the simulation from time zero several times within one run of simulation programme.

The statistical analysis of the system behaviour requires a very large number of simulation repetition, therefore the time performance of developed simulator is very important.

6. Analysis Results of DTS Simulation

6.1. Exemplar DTS

We propose for the case study analysis an exemplar DTS based on Polish Post regional centre in Wroclaw. We have modelled a system consisting of one central node (Wroclaw regional centre) and twenty two other nodes - cities where there are local post distribution points in Dolny Slask Province. The length of roads was a set according to real road distances between cities used in the analysed case study. The intensity of generation of containers for all destinations was a set to 4.16 per hour in each direction giving in average 4400 containers to be transported each day. The vehicles speed was modelled by Gaussian distribution with 50 km/h of mean value and 5 km/h of standard deviation. The average loading time was equal to 5 minutes. There were two types of vehicles: with capacity of 10 and 15 containers. The MTF of each vehicle was set to 2000. The average repair time was set to 5h (Gaussian distribution). The time-table consists of 184 routes. [8]

6.2. Availability metric

The simulation time was set to 100 days and each simulation was repeated 10.000 times. We have calculated functional metrics defined in section 4. The achieved results, functional availability and average one for guaranteed delay 24 and acceptable delay 10 containers, are presented on Figure 1. The functional availability is dropping down each 24 hours due to a definition of time-table, vehicles are not working at night, but containers are entering the system all the time.

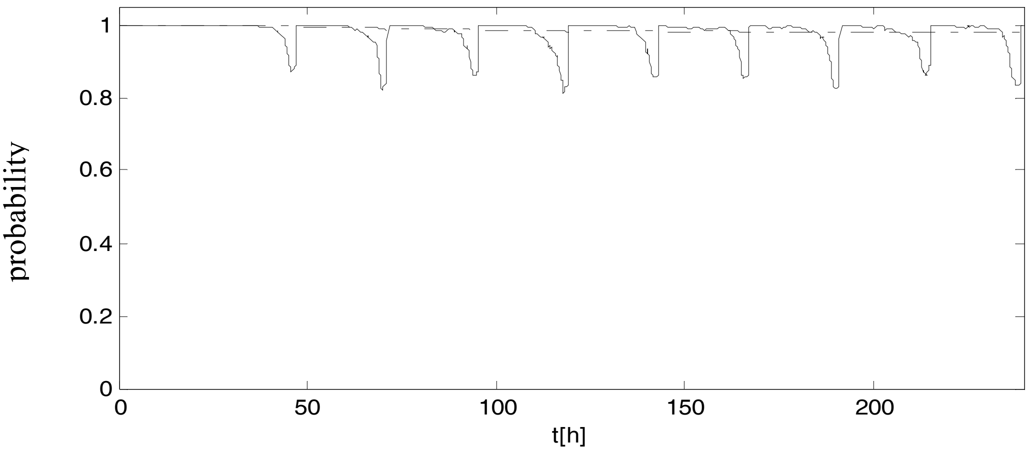
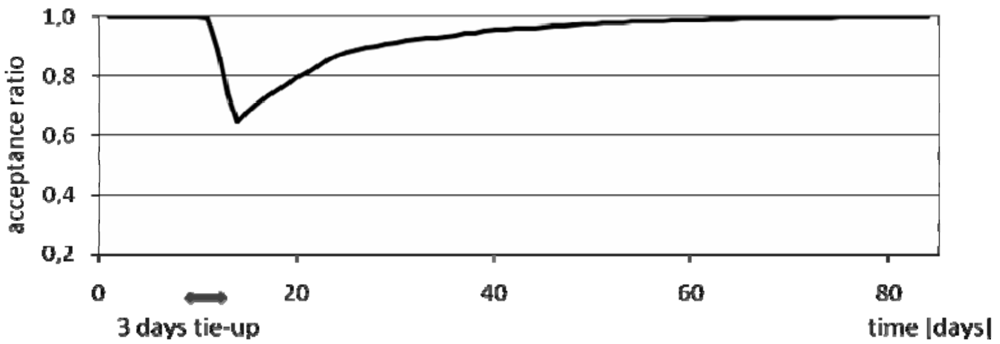


Figure 1. Availability (solid line) and average availability (dashed-dot line) for the exemplar DTS

6.3. Critical situation

The developed simulation allows analysing the transport system performance in case of some critical situations. Let assume that for some days the system is not working at all. The tie-up of the system could be caused for example by a driver strike. After a given number of days the system is again working. The achieved results (acceptance ratio calculated according to (17)) for 3 and 10 days tie-up are presented on Figure 2a and 2. As it could be expected the acceptance ratio in day 6 (when tie-up starts) is starting to drop down and when drivers come back is slowly enlarging. The Figure 3 presents how many days are needed for the transport system to achieve a required level (0.9, 0.95 and 0.98) after a tie-up of the different length.

a)



b)

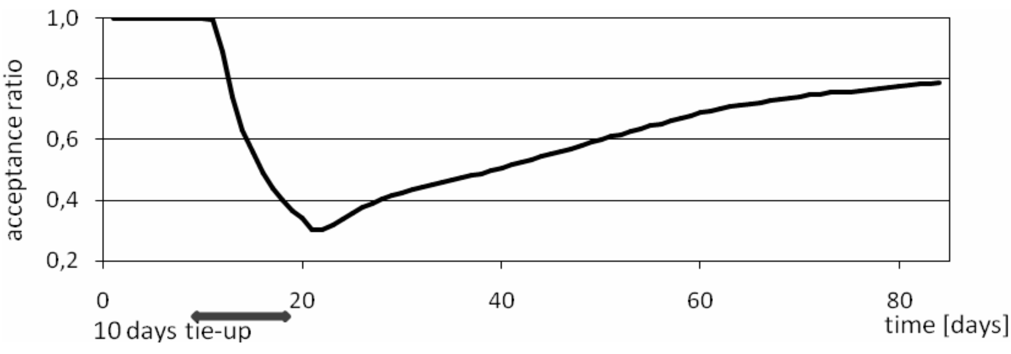


Figure 2. Acceptance ratio for a 3 days (a) and 10 days (b) tie-up

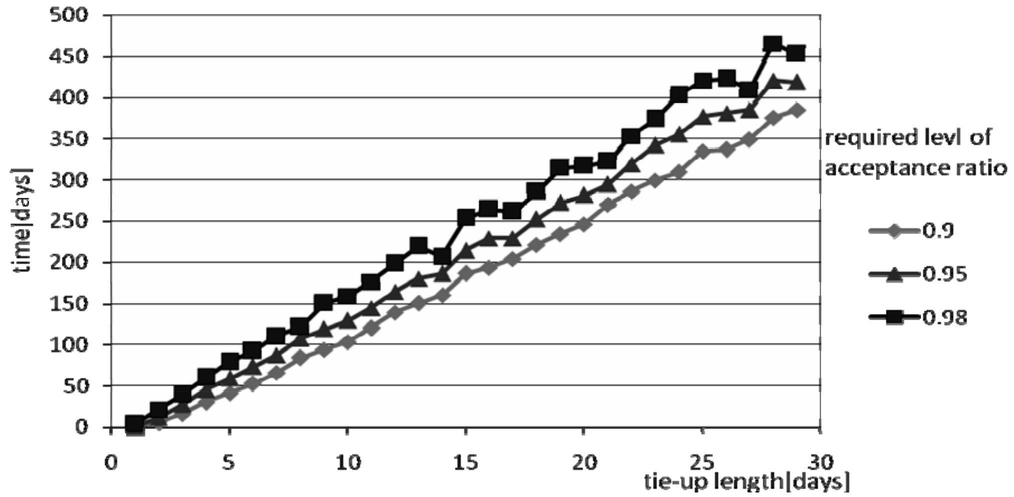


Figure 3. The time required to achieve a given level of acceptance ratio after a tie-up of different length for the exemplar DTS

6.4. DTS simulator performance

Next, we have tested the DTS simulator performance and scalability. We calculated the time of running one batch of simulation of the exemplar DTS presented above for 100 days on a 2.80 GHz Intel Core Duo machine on Linux system (Table 1). The CYGWIN base Windows implementation of PRIME SSF was around 10 times slower then Linux one. Next, we have enlarged the number of containers transported each day 10, 50, 100 and 500 times and proportionally enlarged the transport system (number of trucks, routes and service points). As it could be seen in Table 1 and Figure 4 the memory usage is linearly proportional to a number of containers transported each day, whereas the simulation time is polynomial proportional.

We think that the time and memory effectiveness of simulation done in PRIME environment is very promising. Of course the time needed to perform one simulation depends on the number of events presented in the system, which is a result DTS configuration.

Table 1. DTS simulator performance

	Reference DTS	10 x larger	50 x larger	100 x larger	500 x larger
Number of trucks	52	520	2 600	5 200	26 000
Number of router per day	184	1 840	9 200	18 400	92 000
Number of containers per day	4 400	44 000	220 000	440 000	2 200 000
Simulation time	0.21s	2.60 s	31.91 s	154.5 s	1910.0 s
Memory usage	2 MB	12 MB	63 MB	125 MB	600 MB

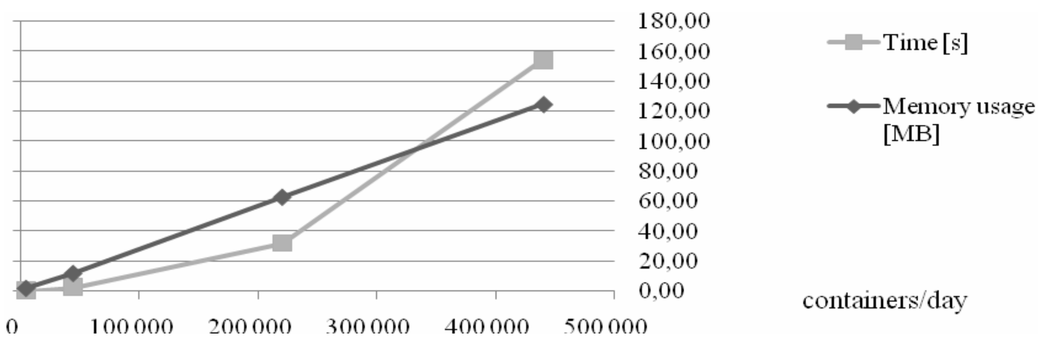


Figure 4. Simulator simulation time and memory usage in a function of number of containers per day generated in DTS

7. Conclusions

We have presented a formal model and event-driven simulator of discrete transport system (DTS). The DTS model is based on Polish Post regional transport system. Simulator was implemented using the Scalable Simulation Framework (SSF). The simulator allows performing reliability and functional analysis of the DTS, for example:

- determine what will cause a ”local” change in the system,
- make experiments in case of increasing number of containers per day incoming to system,
- identify weak point of the system by comparing few its configuration,
- better understand how the system behaves.

Based on the results of simulation it is possible to create different metrics to analyse the system in case of reliability, functional and economic case. The availability, average availability and acceptance ratio of the system was introduced – defined in a functional way by delayed tasks realization. The metric could be analysed as a function of different essential functional and reliability parameters of DTS. Also the system could be analysed in case of some critical situation (like, for example, a few day tie-up). The paper includes some exemplar systems, based on real Polish Post Wroclaw area, and calculated metric.

The achieved performance of the DTS simulator makes it a practical tool for defining an organization of vehicle maintenance and transport system logistics.

References

1. Sanso, B., Milot, L. Perform ability of a Congested Urban-Transportation Network when Accident Information is Available, *Transportation Science*, Vol. 33, No 1, 1999.
2. Gold, N., Knight, C. , Mohan, A. and Munro, M. Understanding service-oriented software, *IEEE Software*, Vol. 21, 2004, pp. 71-77.
3. Birta, L., Arbez, G. *Modelling and Simulation: Exploring Dynamic System Behaviour*. London: Springer, 2007.
4. Fishman, G. *Monte Carlo: Concepts, Algorithms, and Applications*. New York: Springer-Verlag, 1996.
5. Barlow, R., Proschan, F. *Mathematical Theory of Reliability*, Philadelphia: Society for Industrial and Applied Mathematics, 1996.
6. Walkowiak, T., Mazurkiewicz, J. Functional Availability Analysis of Discrete Transport System Realized by SSF Simulator. In: *Computational Science – ICCS 2008, 8th International Conference, Krakow, Poland, June 2008, part I, LNCS 5101* Berlin, Heidelberg: Springer-Verlag, 2008, pp. 671-678.
7. Walkowiak, T., Mazurkiewicz, J. Availability of Discrete Transport System Simulated by SSF Tool. In: *Proceedings of International Conference on Dependability of Computer Systems*, *Szklarska Poreba*, *Poland, June 2008*. Los Alamitos: IEEE Computer Society Press, 2008, pp. 430-437.
8. Walkowiak, T., Mazurkieiwicz, J. Analysis of critical situations in discrete transport systems. In : *Proceedings of International Conference on Dependability of Computer Systems*, *Brunow*, *Poland, June 30-July 2, 2009*. Los Alamitos: IEEE Computer Society Press, 2009, pp. 364-371.
9. Nicol, D., Liu, J., Liljenstam, M., Guanhua, Y. Simulation of large scale networks using SSF. In: *Proceedings of the 2003 Winter Simulation Conference, vol. 1,* New Orleans, 2000, pp. 650-657.
10. Liu, J. *Parallel Real-time Immersive Modeling Environment (PRIME), Scalable Simulation Framework (SSF): User’s manual*. Colorado School of Mines Department of Mathematical and Computer Sciences, 2006. [Online]. Available: http://prime.mines.edu/.