Mazurkiewicz Jacek
Wrocław University of Technology, Poland

Dependability of discrete transport system - model, simulation, measures

Keywords

discrete transport system, system dependability, functional and dependability models, Monte-Carlo simulation

Abstract

The paper proposes a method of reliability and functional analysis related to discrete transport systems. The proposed analysis is based on modelling and simulating of the system behaviour. Monte Carlo simulation is used for proper reliability and functional parameters calculation. The simulator is built using Scalable Simulation Framework (*SSF*). No restriction on the system structure and on a kind of distribution is the main advantage of the method. The paper presents some exemplar system modelling. The authors stress the problem of influence of the reliability parameters for final functional measures (required time of delivery) – the key value to calculate the availability of the system. They also propose to measure the economic quality of discrete transport system by "profit function". The presented problem is practically essential for defining an organization of vehicle maintenance and transportation system logistics.

1. Introduction

Decisions related to technical systems ought to be taken based on different and sometimes contradictory conditions. The reliability maybe is not the most important factor but is of a great weight as a support criterion. So quantitative information related to the reliability characteristics is important and can be used to create a decision-aided system if it is necessary to discuss different economic aspects. But reliability criteria calculation is not trivial in general and especially in a case of modern transportation systems which often have a complex network of connections. From the reliability point of view [1] the transportation systems are characterized by a very complex structure. The performance of the network can be impaired by various types of faults related to the transport vehicles, communication infrastructure or even by traffic congestion [8]. This analysis can only be done if there is a formal model of the transport logistics. The classical models used for reliability analysis are mainly based on Markov or Semi-Markov processes [2] which are idealized and it is hard to reconcile them with practice. We suggest the Monte Carlo simulation [7] for proper reliability and functional parameters calculation. No restriction on the system structure and on a kind of distribution is the main advantage of the method [16]. We propose to use the *SSF* (Scalable Simulation Framework) [17] instead of dedicated system elaboration. Our previous works perfectly show that is very hard to prepare the simulator which includes all aspects of discrete transport. The *SSF* is a base for *SSFNet* [17] a popular simulator of computer networks. We developed an extension to *SSF* allowing to simulate transportation systems. We propose the formal model of discrete transport system to analyze functional aspects of complex systems.

The analysis is based on the real transportation system of the Polish Post – described in the next section. A base for the discrete transport system model is introduced in a section 3. The management problems are discussed in the section 5. The main service given by the post system is the delivery of mails. From the client 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 performance of the analysed system is measured by the availability defined as a ratio of delivered mail on time (defined in section 6). Calculation of the availability is done inside the simulator of the discrete transport system presented in a section 4. Next (section 7), we give an example

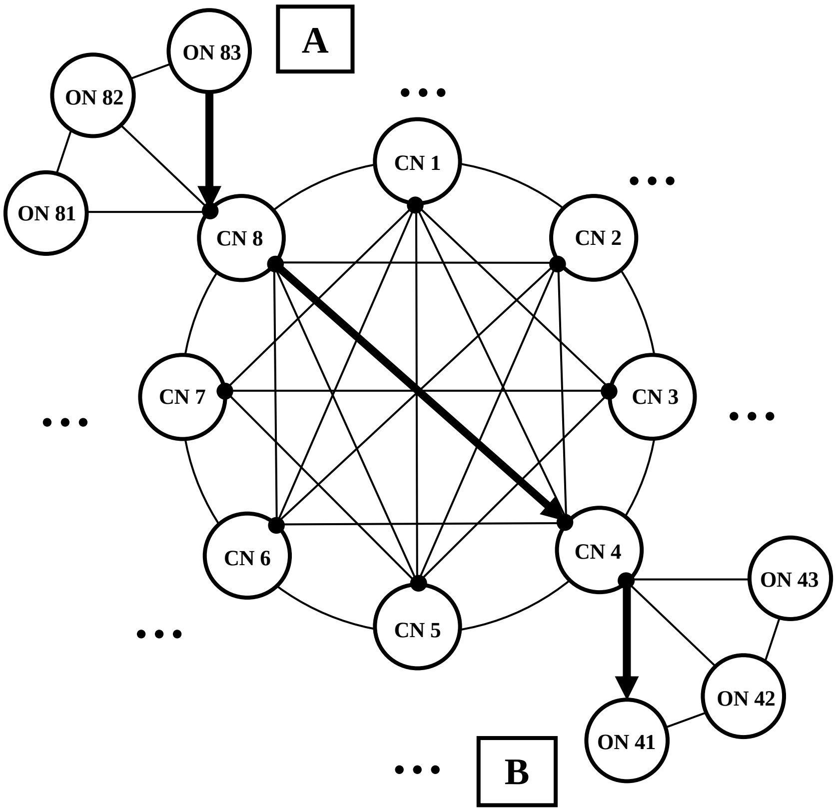


Figure 1. The general structure of the Polish Post transportation system

of using presented model and simulator for the analysis of the Polish Post regional centre in Wroclaw transportation system.

2. Discrete transport system with central node

The analyzed transportation system is a simplified case of the Polish Post. The overall system structure is presented in *Figure 1*. The business service [17], [18], [19] provided the Polish Post is the delivery of mails. The system consists of a set of nodes (marked by circles in *Figure 1*) 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 marked by lines in Figure 1. 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* (see *Figure 1*). At first the mail is transported to the nearest ordinary node *A*. In our example it is marked as *ON83*. 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 (*CN8* in our example). In central node containers are repacked and delivered to appropriate (according to delivery address of each mail) central node. In our example the mail has to be delivered to point *B* therefore from *CN8* is transported *CN4*. In the second (first was *CN8* the second *CN4*) central node the mail is again repacked and deliver in a container to destination ordinary node (*ON41*).

Therefore, in our example the mail was transported throughout 4 nodes: *ON83*, *CN8*, *CN4* and *ON41*.

In the Polish Post there are 14 central nodes and more than 300 ordinary nodes. There more than one million mails going through one central node within 24 hours. It gives a very large system to be modeled and simulated. Therefore, we have decided only a part of the Polish Post transportation system – one central node with a set of ordinary nodes.

The process of any system modeling requires to define the level of details. Increasing the system 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 modeling could not allow to record required data for system measure calculation. Therefore, the level of system model details should be defined by requirements of the system measure calculation. Since number of mails presented in the modeled 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.

3. Discrete transport system formal model

The model can be described as follows:

$$DTS = \langle CN, N, R, V, T, M, TT \rangle \quad (1)$$

where:
CN – central node,
N – set of ordinary nodes,
R – set of routes,
V – set of vehicles,
T – set of tasks,
M – set of maintenance crews,
TT – vehicles' time-table.

Commodities: We can discuss several kinds of a commodity transported in the system. Single kind commodity is placed in a unified container, and containers are transported by vehicles. The commodities are addressed and there are no other parameters describing them.

Nodes: We have single central node in the system. The central node is the destination of all commodities taken from other – ordinary nodes. The central node is also the global generator of commodities driven to the nodes of the system. The generation of containers is described by Poisson 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 other direction.

Ordinary nodes are described by intensity of container generation (routed to central node) and central node is described be a table of intensities of containers for each ordinary node. Moreover the length between each two nodes is given.

Vehicles: We assumed that all vehicles are of the same type and are described by following functional and reliability parameters: mean speed of a journey, capacity – number of containers which can be loaded, reliability function and time of vehicle maintenance. Central node is the base place for vehicles. They start from the central node and the central node is the destination of their travel. The temporary state of each vehicle is characterized by following data: vehicle state, distance travelled from the begin of the route, capacity of the commodity. The vehicle running to the end of the route is able to take different kinds of commodity (located in unified containers, each container includes single-kind commodity).
The vehicle hauling a commodity is always fully loaded or taking the last part of the commodity if it is less than its capacity.
Routes: Each route describes possible trip of vehicles. The set of routes we can describe as series of nodes:

$$R = \langle c, v_1, \ldots, v_n, c \rangle \quad v_i \in N \qquad (2)$$
$$and \quad c = CN$$

Maintenance Crews: Maintenance crews are identical and indistinguishable. 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 temporary state of maintenance crew is characterized by: number of crews which are not involved into maintenance procedures and queue of vehicle waiting for the maintenance.

Time-Table: Vehicles operate according to the time-table exactly as city buses or intercity coaches. The time-table consists of a set of routes (sequence of nodes starting and ending in the central node, times of approaching each node in the route and the recommended size of a vehicle.

The number of used vehicles or the capacity of vehicles does 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 journey 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.
Each day a given time-table is realized, it means that at a time given by the time table a vehicle, selected randomly 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. Next, after approaching given node (it takes some time according to vehicle speed - random process and road length) and the vehicle 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 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. The operation is repeated in each node on the

route and finally the vehicle is approaching the central node when 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.

4. Discrete transport system simulation

4.1. Event-driven simulation

Discrete transport system described in the section 3 is very hard to be analysed by formal methods. It does not lay in the Markov process framework [2]. 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 so called event-driven simulation. Which is based on an idea of event, which could is 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 [7], what requires a large number of repeated simulations.

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

- beginning state of a *DTS* initialization,
- event state initialisation, set time $t = 0$,
- repeat until $t < \tau$:
- 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.

4.2. 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 is done in objects representing *DTS* elements. The objects are working in parallel. Following types of system elements were distinguished: vehicle, ordinary node, central node, time table [16].

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 is 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 has additional process (working in parallel) which is 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 [17].

4.3. 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*) [16] which is a used for *SSFNet* [16], [17] 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*) [17] 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.

5. Fleet management discussion

5.1. Traffic modelling problem

Modelling traffic flow for design, planning and management of transportation systems in urban and highway area has been addressed since the 1950s mostly by the civil engineering community. The following definitions and concepts of traffic simulation modelling can be found in works such as Gartner et al. [8]. Depending on the level of detail in modelling the granularity of traffic flow, traffic models are broadly divided into two categories: macroscopic and microscopic models. According to Gartner et al. [8], a macroscopic model describes the traffic flow as a fluid process with aggregate variables, such as flow and density. The state of the system is then simulated using analytical relationships between average variables such as traffic density, traffic volume, and average speed. On the other hand, a microscopic model reproduces interaction of punctual elements (vehicles, road segments, intersections, etc) in the traffic network. Each vehicle in the system is emulated according to its individual characteristics (length, speed, acceleration, etc.). Traffic is then simulated, using processing logic and models describing vehicle driving behaviour, such as car-following and lane-changing models. Those models reproduce driver-driver and driver-road interactions. Despite its great accuracy level, for many years this highly detailed modelling was considered a computationally intensive approach. Since the last twenty years, with the improvements in processing speed, this microscopic approach becomes more attractive. In fact, Ben-Akiva et al. [3], Barcelo et al. [1] and Liu et al. [12] claim that using microscopic approach is essential to track the real-time traffic state and then, to define strategy to decrease congestion in urban transportation networks. For the control of congestion, they explain that the models must accurately capture the full dynamics of time dependant traffic phenomena and must also track vehicles' reactions when exposed to Intelligent Transportation Systems (ITS). From the latter assertions, in order to control traffic congestion in internal transportation networks it appears that the microscopic modelling will be more appropriate. A common definition of congestion is the apparition of a delay above the minimum travel time needed to traverse a transportation network. As stated in Taylor et al. [14], this notion is context-specific; and complex because a delay may always appear in dynamic transportation system, but this delay must exceed a threshold value in order to be considered.

5.2. Microscopic analysis

Few works have considered the traffic behaviour when studying outdoors vehicle-based internal transport operational problems. In the surface mining environment, pickup and delivery operations involve a fleet of trucks transporting materials from excavation stations to dumping stations, through a designed shared road network. At pickup stations, shovels are continuously digging during a shift according to a pre-assigned mining production plan. Trucks are moving in a cyclical manner between shovels (pickup stations), and dumping areas (delivery stations). A truck cycle time is defined as the time spent by a truck to accomplish an affected mission that consists of travelling to a specific shovel, being serviced by the shovel and hauling material to a specific dumping area. Burt and Caccetta [5] state that mine productivity is very sensitive to truck dispatching decisions which are closely related to the truck cycle time. Thus several papers have studied and proposed algorithms and software to resolve this problematic issue. In fact, this critical decision consists of finding, according to the real environment, to which best shovel a truck must be affected. Such decision has to be generated continuously during a shift, whenever a truck finished dumping at a delivery station. Despite the several proposed dispatching software, recent articles by Krzyzanowska [11] formally criticize the simplistic assumption behind those software which tend to provide dispatching decisions with the objective to optimise a truck cycle times previously calculated. Generally speaking, those software systems based the optimisation process on the past period collected data of trucks cycle times and assume that for the next period trucks will spend on average the same time to accomplish missions. But in the reality of mining operation, the duration of truck travel time appears to be very sensitive to the variable traffic state and road conditions. Burt and Caccetta [5] and Krzyzanowska [11], point out the unresolved problematic of truck bunching and platoon formation in mining road network which apparently induce lower productivity.

5.3. Container movement

Similarly to material transportation in mining operation, several papers (Ioannou [10], Vis [15]) have provided methods for improving container terminal complex operations. In such applications, three types of handling operations are defined: vessel operations, receiving/delivery operations and container handling and storage operations in the stack yards. As we are interested by internal transportation systems, our review concerns the papers dealing with the container handling and storage operations in the stack yards. Generally speaking, vessels bring inbound containers to be picked up by internal trucks and distributed to the respective stocks in the yard. Once discharged, vessels have to leave with on board outbound containers which also are delivered by internal trucks from the storage yard. For this purpose, trucks are moving through a terminal internal road network. In order to decrease the vessel turnaround time, which is the most important performance measure of container terminals, it is important to perform those operations as quickly as possible. In fact according to [3], this movement of containers between quay sides and storage yards appears to greatly affect the productivity of containership’s journey. Vis and Koster [15] gives an extended review of numerous research papers, providing algorithms to solve this complex routing and scheduling problem. They criticize the lack of consistency of the simplistic assumptions made to solve the proposed models within the real-world highly stochastic environment. The ignored traffic situation in the complex seaport internal transportation network is strongly criticized in recent papers [4], [10]. For example, in [3], a travel time of a container internal truck is modelled as a static mean time of travel, based on the distance and the truck average speed. Duinkerken et al. [6], put a uniform distribution between zero and 30% of the nominal travel time formulation, aiming to assimilate the complexity of traffic. More accurate work to solve this issue is the one provided recently by Liu, Chu and Recker [12]. They integrate a traffic model to the internal service model and reported the effectiveness of this integration which allows analysing the tractor traffic flow in a port container terminal. Conscious about the critical problem of congestion in the road network inside a terminal, a quantitative measure of congestion to be added as a controllable decision variable had been developed. For this purpose, they considered the road system inside the terminal as a directed network and they measured flows on arcs in units of trucks travelling per unit time. Those two last works appear as providing the leader approach in term of consideration of congestion and traffic in container terminals; however, their approach is ultimately macroscopic. As we have lately discussed, even if this macroscopic approach allows analysing the traffic behaviour, the highly detailed microscopic model is more efficient for an effective real-time traffic monitoring and control.

6. Functional metrics of DTS

We define the availability of the system as an ability to realize the transportation task within a required time. The availability is defined as probability. Let’s use the following values:

- T – 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 – guaranteed time of delivery, if exceeded the container is delayed.

In *Figure 1*. we can observe two possible situations:
(a) - delivery was realized before guaranteed time T_g,
(b) - delivery was delayed - time of delay: $T - T_g$.
If $N(t)$ is a stochastic process describing the number of delayed containers at time t, the 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.

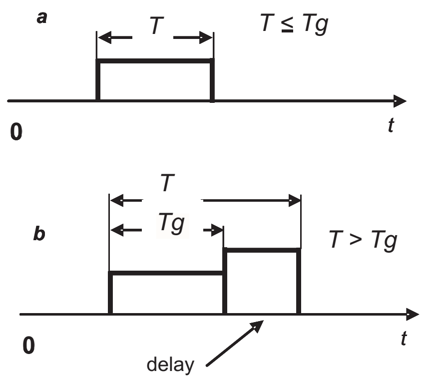


Figure 2. The delivery in guaranteed time (a) and delayed delivery (b)

$$A_k(t) = \boldsymbol{Pr}\{N(t) \leq k\} \tag{3}$$

The availability describes a state of an analyzed system at a given point of time. In case if somebody wants to analyze a state of system in a time interval we propose other metric: average availability $AA_k(t)$. 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):

$$AA_k(t) = \frac{1}{t}\int_0^t \Pr\{N(\tau) \leq k\} d\tau \tag{4}$$

6.1. Economic analysis

The simulation approach to transportation system analysis allows performing economic analysis. We propose to measure the economic quality of discrete transport system by "profit function" $P(DTS_i)$ estimated in given time-period T as (5)
where:
DTS_0 - some reference discrete transport system,
$L(DTS_i)$ - penalties in analyzed time-period, a function of container delays for a given transport system DTS_i,
$c(DTS_i)$ - the costs of one vehicle in analyzed time-period (operating and leasing costs) for a given transport system DTS_i,
$n(DTS_i)$ - number of vehicles in DTS_i.

The profit function defined by (5) is a kind of normalized and relative gain. It could be understood as a possible gain or loss caused by changes in transportation system organization (changes to some reference *DTS*) in one vehicle cost units. The change could be for example caused by increasing or decreasing of number of vehicles.

7. Case study

We proposed 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 ordinary nodes - cities where there are local post distribution points in Dolny Slask Province. The lengths of roads were set according to real road distances between cities used in the analyzed case study. The intensity of generation of containers for all destinations were 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 50km/h of mean value and 5km/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. There was one maintain crew, the average repair time was set to 5h (Gaussian distribution).
To transport all generated in the system containers time-table has to be set-up. For a described example it was designed manually in a manner to allow transporting all containers (with 10% overhead) and 15 minutes of break between each drive. It resulted in 180 drives each day with trucks of size 10 and 15. The last parameter to set-up was a number of trucks of a size 10 and 15. At first it was set to a minimum size that comes from the given time-table. In our case it was 13 trucks of size 10 and 21 of size 15. The availability of the system (4) and average availability (5) was calculated with guaranteed time T_g equal to 24 h and threshold k equal to 10 based on 10000 repeated simulations. In each simulation 40 days were analyzed. Results presented in *Figure 3a* shows that assumed number of trucks is insufficient since the availability drops to 0 after 6 days. It is caused by failures of trucks, delays (due to simulation of traffic jams) and first all of due to the truck drives algorithm i.e. the drive is performed only if any truck is availability at the starting time in the central node.
Now, one could rise a question what should be a number of trucks to have availability equal to 1 in most cases. It is a non trivial question and almost impossible to be answered by analytical methods. However, simulating approach allows to do it, by increasing number of vehicles, repeating simulation for a new system and observing availability metric. The Fig. 3b. and Fig. 3c. show results for enlarged number of trucks. In case of a usage of 19 trucks of size 10 and 27 of size 15 (*Figure. 3c.*) the average availability is over 0.99 in the analysed period time. The system with only two less trucks (*Figure 3b.*) has much worse values of analyzed metrics. It shows how important is to set-up a proper value of number of trucks. It could be also noticed that the availability

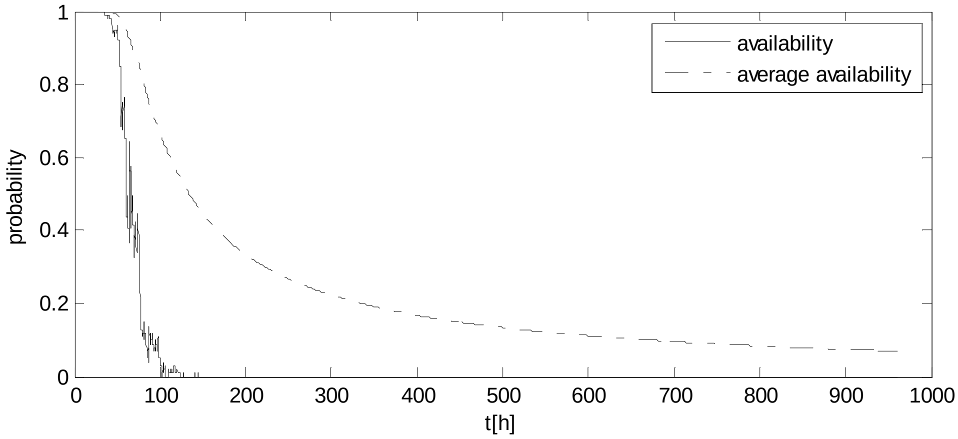
has periodic changes (*Figure 2a-c*. solid line). The situation is an effect of used time-tables and of a method of containers'' generation. The containers are generated during all day (by Poisson process) but according to a time-table trucks do not operate in the night. Therefore, the probability of delay increases at the night.

8. Conclusion

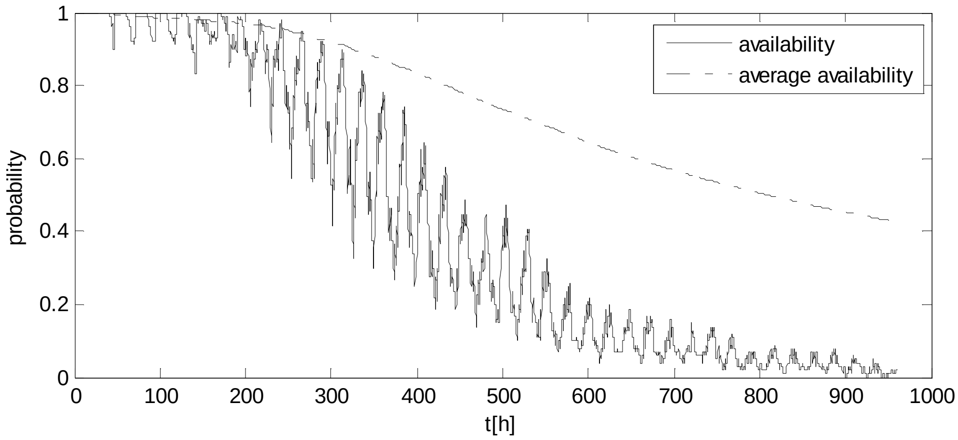
We have presented a simulation approach to functional analysis of Discrete Transport System with Central Node. The *DTS* models the Polish Post regional centre of mail distribution behaviour.

$$P(DTS_i) = \frac{L(DTS_i) - L(DTS_0) + c(DTS_i)\cdot n(DTS_i) - c(DTS_0)\cdot n(DTS_0)}{c(DTS_0)} \quad (5)$$

(a)



(b)



(c)

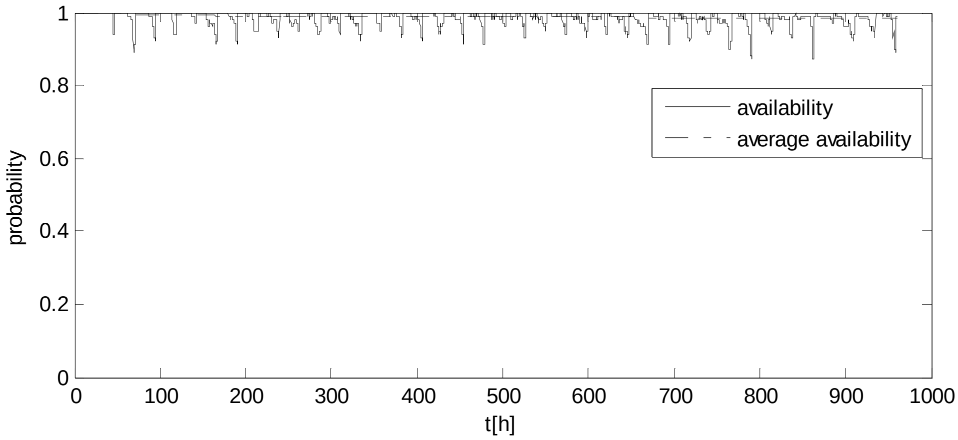


Figure 3. Availability (solid line) and average availability (dashed-dot line) for the exemplar *DTS* for number of trucks equal to: (a) 13 - size 10 and 21 - size 15 (b) 18 and 26 (c) 19 and 27

Developed simulation software allows analyzing different metrics of the system in a function of all model parameters, like for example changes in a management approach or in a number of used trucks. The presented results show results of availability and average availability in a function of a number of used trucks. However, other metrics could be also used. The results of presented experiments could be used for example for selection of the optimum value for SLA (service level agreement). The implementation of *DTS* simulator done based on *SSF* allows applying in a simple and fast way changes in the transportation system model. Also the time performance of *SSF* kernel results in a very effective simulator of discrete transport system. The case study exemplar models a Wroclaw regional post centre with over 180 drives per day. The calculation of one simulation of 40 days requires around 30 seconds on Pentium Centrino 1,7 GHz. Also the memory requirements are not large - around 10 megabytes). Results presented in [18] for a much smaller system (only 5 trucks with 18 drives per day) with around 6 seconds shows very good scalability of proposed approach. Therefore, we think, that introduced exemplar analysis shows, that the described method of transportation system modelling can serve for practical solving of essential decision problems related to an organization and parameters of a real transportation system. The proposed analysis seems to be very useful for mail distribution centre organisation.

Acknowledgment

Work reported in this paper was sponsored by a grant No. N N509 496238, (years: 2010-2013) from the Polish Ministry of Science and Higher Education.

References

[1] Barcelo, J., Codina, E., Casas, J., Ferrer, J.L. & Garcia, D. (2005). Microscopic Traffic Simulation: a Tool for the Design, Analysis And Evaluation Of Intelligent Transport Systems. *Journal of Intelligent and Robotic Systems: Theory and Applications*, Vol. 41, 173-203.

[2] Barlow, R. & Proschan, F. (1996). *Mathematical Theory of Reliability*. Philadelphia: Society for Industrial and Applied Mathematics.

[3] Ben-Akiva, M., Cuneo, D., Hasan, M., Jha, M. & Yang, Q. (2003). Evaluation of Freeway Control Using a Microscopic Simulation Laboratory. *Transportation Research, Part C (Emerging Technologies)*, Vol. 11C, 29-50.

[4] Birta, L. & Arbez, G. (2007). *Modelling and Simulation: Exploring Dynamic System Behaviour*. London: Springer.

[5] Burt, C.N. & Caccetta, L. (2007). Match Factor for Heterogeneous Truck and Loader Fleets. *International Journal of Mining, Reclamation and Environment*, Vol. 21, 262-270.

[6] Duinkerken, M.B., Dekker, R., Kurstjens, S.T.G.L., Ottjes, J.A., & Dellaert, N.P. (2006). Comparing Transportation Systems for Inter-Terminal Transport at the Maasvlakte Container Terminals. *OR Spectrum*, Vol. 28, 469-493.

[7] Fishman, G. (1996). *Monte Carlo: Concepts, Algorithms, and Applications*. Springer-Verlag.

[8] Gartner, N., Messer, C.J. & Rathi, A.K. (1998). *Traffic Flow Theory and Characteristics*. In: T.R. Board (Ed.). Texas: University of Texas at Austin.

[9] Gold, N., Knight, C., Mohan, A. & Munro, M. (2004). Understanding service-oriented software. *IEEE Software*, Vol. 21, 71-77.

[10] Ioannou, P.A. (2008). *Intelligent Freight Transportation.* Carolina: Taylor and Francis Group.

[11] Krzyzanowska, J. (2007). The Impact of Mixed Fleet Hauling on Mining Operations at Venetia Mine. *Journal of The South African Institute of Mining and Metallurgy,* Vol. 107, 215-224.

[12] Liu, H., Chu, L. & Recker, W. (2004). Performance Evaluation of ITS Strategies Using Microscopic Simulation. *Proc. of the 7th International IEEE Conference on Intelligent Transportation Systems*, 255-270.

[13] Sanso, B. & Milot, L. (1999). Performability of a Congested Urban-Transportation Network when Accident Information is Available. *Transportation Science,* Vol. 33, No 1, 10-21.

[14] Taylor, M.A.P., Woolley, J.E. & Zito, R. (2000). Integration of the Global Positioning System and Geographical Information Systems for Traffic Congestion Studies. *Transportation Research, Part C (Emerging Technologies),* Vol. 8C, 257-285.

[15] Vis, I.F.A. (2006). Survey of Research in the Design and Control of Automated Guided Vehicle Systems. *European Journal of Operational Research,* Vol. 170, 677-709.

[16] Walkowiak, T. & Mazurkiewicz, J. (2009). Analysis of Critical Situations in Discrete Transport Systems. International Conference on Dependability of Computer Systems, *IEEE Computer Society Press,* 364-371.

[17] Walkowiak, T. & Mazurkiewicz, J. (2008). Availability of Discrete Transport System Simulated by SSF Tool. International Conference on Dependability of Computer Systems, *IEEE Computer Society Press,* 430-437.

[18] Walkowiak, T. & Mazurkiewicz, J. (2008). Functional Availability Analysis of Discrete Transport System Realized by SSF Simulator. *Computational Science – ICCS 2008,* Springer-Verlag, LNCS 5101, Part I, 671-678.

[19] Walkowiak, T. & Mazurkiewicz, J. (2010). Algorithmic Approach to Vehicle Dispatching in Discrete Transport Systems. *Technical approach to dependability.* Wroclaw, 173-188.

[20] Walkowiak, T. & Mazurkiewicz, J. (2010). Functional Availability Analysis of Discrete Transport System Simulated by SSF Tool. *International Journal of Critical Computer-Based Systems,* Vol. 1, No 1-3, 255-266.

[21] Walkowiak, T. & Mazurkiewicz, J. (2010). *Soft Computing Approach to Discrete Transport System Management.* LNAI 6114, 675-682.