Performability analysis of a tramway system with virtual tags and local positioning

Performability Analysis of a Tramway System with Virtual Tags and Local Positioning

Leandro Dias da Silva*, Diamantea Mongelli†, Paolo Lollini‡§, Andrea Bondavalli‡§, Gianluca Mandò¶
*Computing Institute, Federal University of Alagoas, Maceió, Brazil
Email: leandrodias@ic.ufal.br
†Resiltech S.R.L., Pontedera, Italy
Email: diamantea.mongelli@resiltech.com
‡Dept. of Math. and Computer Science, University of Florence, Firenze, Italy
Email: {lollini,bondavalli}@unifi.it
§Consorzio Interuniversitario Nazionale per l'Informatica (CINI), University of Florence, Firenze, Italy
¶Innovation & Technology, Thales Italia SpA
Email: gianluca.mando@thalesgroup.com

***Abstract*—Current solutions for tramway Interlocking Systems are based on physical sensors (balizes) distributed along the infrastructure which detect passing of the trams and trigger different actions. This approach is not easily scalable and maintainable, and it is costly. The Regional Project SISTER aims at designing new architectural solutions for addressing the previous problems based on the virtualization of the sensors and on the local positioning of each tram. The idea is to trigger actions when the computed local position corresponds to a virtual tag. However, the computed position can be affected by errors, compared to the real one. Therefore, it is important to understand the impact of these new solutions on the performability of the system. This paper focuses on the analysis of the performability of a tramway system, based on the SISTER architectural solutions. We build a model using Stochastic Activity Networks and run sensitivity analyses on i) the accuracy of the positioning, ii) the different SISTER parameters (e.g., those triggering the activation of manual procedures). This analysis allows us to properly set and fine-tune the key architectural parameters, to understand the impact of the accuracy on the positioning, and to understand the impact of failures.**

***Index Terms*—Performability Analysis, Stochastic Activity Networks, Tramway, Interlocking System**

I. Introduction

The SISTER project stems from the idea of defining a solution that addresses the issue of safety in a rail-tram transport system through the development, integration, and validation of innovative signaling, radio communication, and remote sensing technologies.

The innovative architectural solutions developed in SISTER (REGIONE TOSCANA POR FESR 2014-2020 SISTER “SIgnaling & Sensing Technologies in Railway application”[1]) are mainly based on the virtualization of physical sensors and on the local positioning system of each tram. Traditional physical sensors (balizes) used for tram passing detection and for triggering the communications with the ground infrastructure (including the interlocking system) are virtualized, and the actions are triggered when the computed local position corresponds to a virtual tag. The resulting system is then more scalable and maintainable w.r.t. traditional tramway systems, and it is less costly. On the other side it is important to analyze the effects of the new architectural solutions on the traffic that can be supported by a tramway network, for properly setting and fine-tuning the key architectural parameters and for assessing the impact of failures that might affect the different architectural components.

In this paper we present a stochastic modeling approach for analyzing the sensitivity of the tramway traffic to key architectural parameters, including the accuracy of the on-board positioning system, the time-out for the activation of manual procedures, and the quality of the communication network. Furthermore, we analyze the impact on the tramway traffic of the level of interference between the trams and the surrounding environment, and of possible failures that can temporarily block the journey of a tram and congest the traffic of the entire tramway.

The tramway domain has been studied in different contexts such as energy consumption [1], attractiveness [2], and multimodal traveler information systems [3]. On the other hand, stochastic models have been applied to several different analysis domains such as failure propagation in chemical plants [4], failure in electric power systems [5], performance and availability of hybrid storage systems [6], and predicting the propagation of train delays [7]. Specifically, in [8], Stochastic Time Petri Nets are used for performability evaluation of emergency stops for the European Rail Traffic Management System. The focus is on communication failures due to connection losses and cell handovers. A sensitivity analysis is presented for different parameters such as train speed, and lost messages. None of the existing works is considering the distinguishing aspects of the SISTER solutions based on the virtualization of the physical sensors.

The analyses presented in this paper are performed by developing a stochastic model based on Stochastic Activity Networks (SAN) [9] that can be simulated using the simulator integrated into the Möbius tool [10], [11]. Models have been developed in a parametric and modular way, so to facilitate

[1]http://www.progetto-sister.com/

their extension and their application to other scenarios and different topologies of the tramway networks.

The paper is organized as it follows. Section II outlines the key SISTER architectural elements. Section III discusses the aspects of the system that affect the metrics of interest and have been captured by the models. The stochastic models are then presented in Section IV, while results are presented and discussed in Section V. Section VI concludes the paper.

II. SISTER MAIN ARCHITECTURAL COMPONENTS

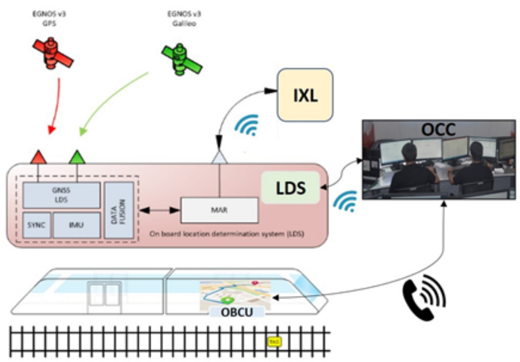

Fig. 1. SISTER system.

Currently, the tram detection is carried out through track circuits and axis counting system. The purpose of the innovative signaling system proposed in SISTER is the use of new tram detection technologies, such as Local Determination System (LDS) based on Global Navigation Satellite System (GNSS) and inertial technologies. The new system for detecting the position of the tram implements a detection system based on the architecture shown in the Figure 1.

Moving from a traditional tram detection system to an LDS architecture, the detection procedure changes from a physical to a virtual one (Virtual Tags). The LDS architecture does not impact on the logic of defining a route, nor on its management by the interlocking that manages an entire tram area.

The main components of the architecture of the new SISTER system are Location Determination System (LDS), Operational Control Center (OCC), On-Board Control Unit (OBCU), and Interlocking (IXL).

Location Determination System (LDS)

The Location Determination System (LDS) is a SIL4 component located at the middle of the tram that computes the position of the tram and communicates it to OCC (Operational Control Center) and to OBCU (On-Board Control Unit on the tram) once per second. LDS calculates the tram position with a maximum error with respect to its real position, and this maximum value represents the **Accuracy** parameter. Therefore, the true position of the tram will be in the range $\pm$ **Accuracy** from the position estimated by the LDS (SIL4 function of LDS).

Operational Control Center (OCC)

The Operational Control Center (OCC) is the ground operation center and can communicate by telephone with the tram driver for the management of certain situations while approaching a junction area. More in detail, if the driver arrives in front of the traffic lights set to STOP, after a specific timeout the driver must contact OCC for understanding what is the problem and for activating a manual procedure.

On-Board Control Unit (OBCU)

The On-Board Control Unit (OBCU) computer is placed on the tram for supporting the driver. OBCU receives the positioning data from LDS once per second. When it receives the data, it compares it with a pre-loaded map that contains three different types of virtual tags: Connection Request, Route Request and Track Circuit Tag. When OBCU detects the first virtual tag (connection request), it sends a connection request message to the IXL. Subsequently, when it detects the second virtual tag (route request) on the map, it sends the route creation request message to the IXL. If the IXL, after appropriate checks, sends the route created message, then the traffic light is set to GO otherwise it is maintained at STOP. In the first case the driver can go ahead and enter in the Track Circuit, while OBCU continues to communicate its position to IXL once per second. The virtual Track Circuit Tag corresponds to a junction area, e.g. corresponding to a tram platform for passengers transfers. IXL correctly receives the positioning data from OBCU during the occupation of the Junction Area, once per second, in order to detect when the track circuit is free. After that, it closes the management of the junction area and clears the route for other requests.

Interlocking (IXL)

The Interlocking (IXL) is an external component that manages a specific tram area and can manage a defined number of trams simultaneously. IXL accepts the connection request received from the On-Board Control Unit (OBCU) of the tram that requested the connection. At receiving the route request from OBCU of the tram, the IXL performs the feasibility check to evaluate if the requested route is available. If it is available then it commands the maneuvering boxes, sets the traffic light signal to GO and sends the route creation message to OBCU.

During the occupation of the junction area, OBCU periodically communicates its positioning data. Therefore, IXL verifies the occupation/release of the virtual Track Circuit, through the location referencing function. When IXL detects that the last virtual track circuit is free from the tram, it sends a disconnection request message to the OBCU, so that it can make it available for other requests.

It is important to note that IXL uses the LDS Accuracy default value **conservatively** for the purpose of occupying virtual Track Circuits (SIL4 functionality of IXL). The Track Circuit is set to occupy as soon as IXL receives information that the tram can *potentially* be found on the virtual TC, which means that it will be occupied Accuracy meters before the tram position communicated by LDS, and it will free the track circuit only when the tram has surely *completely* exited from the last virtual Track Circuit, which means that it will be freed Accuracy meters after the tram position communicated by LDS.

III. Modeling of an operational scenario

In this section we describe the aspects of the SISTER sub-systems that have been captured by stochastic models. The detailed description of the corresponding SAN models is discussed in the next section.

The models have been developed for computing the following performance metrics: Average Trip Time of each tram involved in the scenario; Average number of times the driver had to call OCC and activate the manual procedure. A performability analysis was carried out considering how such metrics will be affected by different levels of the quality offered by the communication networks, and by temporary outages that may affect a tram, with their cascading effects on the following trams. All the model parameters introduced in this section are also reported in Table I.

A. Tram

The main aspects captured by the Tram model are: the topology of the tram network (including virtual tags, traffic lights, and track circuit); the movement of a single tram on the route; the accuracy of the positioning system (LDS component); the drivers call to the OCC; the exchange of messages with IXL (OBCU component), i.e., the connection request, route request, and sending of interlocking disconnection confirmation message (IXL).

The line is composed of discrete segments of fixed length. Each segment can be:

- segment with Connection Request Tag (RC), which represents a segment of the route where the tram detects the first Request Connection Virtual Tag on the OBCU and sends the Connection Request to IXL;
- segment with Route Request Tag (RR), which represents a segment of the route where the tram detects the second Virtual Tag on the OBCU and sends the Route Request to IXL. The distance between the Route Request Tag and the Traffic Light is defined by the parameter $rrtotl$;
- segment with Track Circuit Tag (TC), which represents the segment of the itinerary in which the tram occupies the virtual track circuits placed after the traffic light, which determine the occupation or not of the route. The length of this segment is defined by the $LenTotalCircuits$ parameter;
- an ordinary segment that represents a segment which does not correspond to virtual tags, of length ol.

In the case of a free segment, the tram travels its route at an average speed indicated with the $AverageVelocity$ parameter. Each tram starts its trip from a time established in the definition of the timetable of the line. Trams must maintain a minimum distance between them and cannot occupy the same line segments simultaneously. This also determines that trams enter and exit the line in order. Passenger stop time at the tram platform is defined by the pt parameter. The time taken by the tram to travel an ordinary segment of the line (not corresponding to any virtual tag) is distributed according to a uniform distribution with $LowerBound = (ol/AverageVelocity) * (1 - Bound)$, and $UpperBound = (ol/AverageVelocity) * (1 + Bound)$, where Bound is precisely the percentage of upper and lower bound used in the uniform distribution, and ol is the size of the ordinary segment.

In correspondence with the virtual connection request and route request tags, the Tram sends the respective requests to the interlocking. If the driver arrives in front of the traffic lights to meet the traffic light set as prevented, after a time defined by the $TMax$ system parameter, he must contact the ground operations center (OCC) to know how to proceed. Following the activation of the manual procedure, it is assumed that the tram accumulates a certain delay defined by the $Delay$ parameter, and then continues on its itinerary occupying the segment with the Track Circuit Tag. When IXL detects that the segment with the track circuit is freed from the tram, it sends a message of disconnection request to OBCU that accepts it.

As previously discussed, IXL uses the default LDS Accuracy value conservatively for the purpose of occupying virtual Track Circuits. The model variable that corresponds to the Accuracy value is $Epsilon$. The impact of the Accuracy parameter was then captured in the model as follows. The segment with the Track Circuit Tag is considered occupied by IXL exactly $Epsilon$ meters before the tram has actually occupied it, and it will be freed only $Epsilon$ meters after the tram has actually left the segment.

B. IXL

IXL is the model that captures the functional behavior of the Interlocking, namely the management of a specific area of trams, routes, control boxes, and Traffic Lights. The model has captured the management of connection requests and route requests received from OBCU and the disconnection with it. The IXL model captures the main features of the interlocking system and its communications with the OBCU. In particular, after accepting the connection request received from OBCU and after receiving the route request also from OBCU, IXL performs the feasibility check to verify if the route is available based on the occupation or not of the virtual Track Circuits with respect to the route that was requested. There is just one parameter for this model, $RateIXL$, that corresponds to the average response time to each request.

C. Network

The Network model represents the communication channel for the exchange of messages between Tram (OBCU) and interlocking (IXL). The quality of the communication channel is captured through a parameter p that represents the probability that a single message is not delivered to the recipient, so as to be able to study the quality of communication on the performance of the tramway.

IV. Stochastic Activity Networks (SAN) model

In this section, we describe the technical details of the Stochastic Activity Networks model built for the analysis of the tram network performance considering the specificities of

Parameter	Description	Value
Bound	The percentage of upper and lower bounds of the uniform distribution representing the time taken by the tram to travel an ordinary segment of the line: $LowerBound = (ol/AverageVelocity) * (1 - Bound)$, and $UpperBound = (ol/AverageVelocity) * (1 + Bound)$.	variable
Epsilon	The maximum error on the tram position calculated by the LDS system (Accuracy)	variable
TMax	The maximum time that a tram can wait in front of a traffic light before contacting the OCC	variable
p	Probability that a message does not arrive at its destination	variable
AverageVelocity	Average tram speed	14.0 meters/second
ArrivalTime	Time between the departure of two trams	180 seconds
RateIXL	IXL processing time in response to connection request and route request messages	0.1 seconds
RateNetwork	Time of transmission of a message from OBCU to IXL or from IXL to OBCU	0.08 seconds
Delay	The additional delay accumulated by a tram following the activation of the manual procedure	120 seconds
LenTotalCircuits	The length of segments with Track Circuit Tags	100 meters
nt	The number of trams crossing the line	10
ol	The length of the ordinary segment	50 meters
rrtotl	The distance between the Route Request Tag and the Traffic Light	200 meters
pt	The time at the tram platform	20 seconds

TABLE I

A REAL-LIFE SETTING OF MODEL PARAMETERS.

the SISTER system. For space limitations we had to abstract some parts of the models (mainly the definitions of the input and output gates), but the full technical description is available at the repository[2].

A. Modeling methodology

The model has been developed following modularity and generality criteria. Regarding the modularity, the system has been modeled as the composition of atomic models, thus facilitating both the development and maintenance of the models. In particular, the three atomic models developed are Tram, IXL, and Network.

The Tram model, which characterizes the movement of a single tram in the line, is replicated using the REP (replicate) composition operator, to represent a certain number of trams running along the line. The Tram models replicated in this way share some places that are common to all replicas. The number of replicas to be performed is given by a global variable called nt (number of trams). The replicated model of Tram is then composed with the IXL model and with the Network model using the composition operator JOIN and sharing some places among the different atomic models, thus forming the overall model of the system under examination.

Regarding the generality, the models were created to make them easily applicable to different tramway topologies characterized by different sequences of junction areas. The objective has been achieved through a parametric definition of the data structures used in the model. Further details will be provided during the description of the models.

B. The Tram SAN model

The Tram atomic model is shown in Figure 2. The model will be presented by discussing its sub-parts. The part of the model connected to the output gate OG1 is used to make replicas of the Tram model non-anonymous and to initialize the model with the network topology to be analyzed. As previously mentioned, the Tram model is the model of a single tram. To have several trams at simulation time, this model is replicated through the anonymous replication operator REP.

[2]https://drive.google.com/open?id=1nF3kVXburObq-HOmZ8ZyqXYmGvB9ATkc

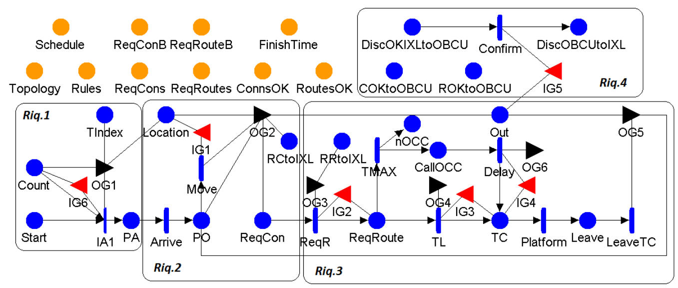

Fig. 2. Tram SAN model.

However, to differentiate the behavior of every single replica (tram) it is necessary to be able to distinguish each replica (tram) through a unique index (identifier). At the beginning of the simulation, the place Start contains one token. Thus, the firing of the IA1 instant activity will take place only once for each replica. When IA1 is triggered, the TIndex place marking is set as nt minus the Count place marking (shared among all replicas), which is then decremented by one token. Also, IA1 firing moves the token into place PA.

The Topology extended place (shared among all replicas) has been defined as an array of size equal to the number of segments (nl) present in the line, where the i-th element indicates the position of each tram along the line. Initially, all the trams are in the first position of the array. That is, all the trams are waiting to enter the scenario according to their scheduled timetable. Thus, the position is updated as the tram moves along the route. The place Location is also set to one and represents the current position of the tram. The IG6 gate plays the fundamental role of initializing the parameters of the model and in particular that of defining the topology of the line and the departure times of the trams.

The specification of the line is made by an array global variable defining the roles of the various segments of the line. As already highlighted, the portion of tramway considered has been discretized into segments with different roles associated with it depending on the type of segment. The extended place Rules is set with the contents of the array and is shared with all replicas. In this way, it is possible to easily define the topology of the line. The only constraint is that the segments with the connection request, route request, and track circuit tags roles must be consecutive, as in the real system. These triple segments can be separated by any number of ordinary segments. The array representing the topology of the line is currently defined in the IG6 input function, but could also be read externally from a configuration file.

The tram scheduling is parametric and is defined by an array with the extended place Schedule, which represents the departure time of each tram. The first tram (index 0) occupies the first position of the array and the others the subsequent positions. The variable at is a variable of type double which represents the average time that elapses between the departure of two consecutive trams (for example every 3 minutes).

The Arrive transition models the beginning of the i-th tram route along the line, following the scheduled time. The Arrive transition firing causes the insertion of a token in the place PO that represents the actual departure of the i-th tram.

The Move activity represents the time taken by the tram to travel an ordinary section of the line (not corresponding to any virtual tag), and has a uniform distribution as can be seen in Table I. The variable ol is a global double variable that represents the length of the ordinary segment. The predicate input of the IG1 input gate models the fact that a tram can only move to the next line section if it is not currently occupied by another tram.

When the Move transition fires, the token is removed from the place PO, and the OG2 gate output code is executed. The output function of the OG2 gate checks whether the tram is along an ordinary segment, and a token is put in the place PO. If the tram is along a segment with Connection Request Tag, a token is inserted in place ReqCon and place RCToIXL. Moreover, the array of the extended place ReqCons is updated, which is an array that keeps track of the trams that forwarded the connection request to IXL, having detected the first Virtual Tag on the OBCU. For each tram, the output function of OG2 also updates the extended place ReqConB, which is a buffer implemented as an array that processes connection requests in FIFO order. Then the Topology extended place is updated, with the (unique) index of the tram that forwarded the connection request, which is then in the i-th position of the segment with Connection Request Tag. It also inserts a token in the place Location.

If places PO and Location do not have tokens, and if the extended place Topology is at position 0, it inserts a token in the place Out and updates the array of the extended place FinishTime, which is an array used to keep track of the end of path times, i.e., the time for each tram leaving the route. $LastActionTime$ is a call available in the simulator which returns the simulated time of the last completed action, i.e., the simulated time in which the last activity fired. This value allows us to calculate the average trip time of each tram.

The ReqR activity represents the crossing time of the segment with Virtual Request Route tag, distributed according to the uniform distribution: Lower Bound: $((rrtotl - epsilon)/AverageVelocity) * 0.90$; Upper bound: $((rrtotl - epsilon)/AverageVelocity) * 1.10$. The variable $rrtotl$ is a global double variable that represents the length of the segment with Route Request Tag, while $epsilon$ is the global double variable that corresponds to the Accuracy parameter of the LDS component. As the value of epsilon increases, the tram will require the route in advance.

The Output function OG3 updates the extended place ReqRoutes, which is an array used to track the trams that require the route to IXL, with the unique index of the tram that forwarded the request, as they detected the second virtual tag on the OBCU. For each tram, it updates the extended place ReqRouteB, which represents a buffer that processes route requests in FIFO order. Moreover, OG3 updates the place Topology with the tram index in the array position that indicates the segment with the Route Request Tag, and increments the Location place marking by one token and set the RRtoIXL marking to one to trace the request to IXL. Furthermore, the firing of the ReqR activity determines the movement of a token in Place ReqRoute. The presence of a token in this Place indicates that the tram is waiting for the route to be created by IXL and that the Traffic Light must be set to GO before it can be overtaken. The input predicate of the IG2 input gate models the fact that a tram can only move to the next line segment if it is not currently occupied by another tram.

The TL (Traffic Light) activity represents the traffic light that is managed by IXL, together with the requests received from the tram and the shunting stations. If the tram does not receive the confirmation message of the creation of the route from IXL or the traffic light is set to red, after a deterministic time indicated by the TMax variable in the TMAX activity, the tram driver must go to the manual procedure to call OCC. In this case, a token is inserted in the nOCC and CallOCC places. The place nOCC is used to calculate the number of times the manual procedure is activated in a certain time interval. This visual procedure adds a delay to the route of the tram on its itinerary, modeled with a deterministic activity Delay. The OG6 output function updates the place Topology, with the tram index in the array position which indicates the segment with Track Circuit and increases the place Location by one token (moves to the next section). The input predicate of the IG4 input gate models the fact that a tram can only move to the next line section if it is not currently occupied by another tram. Otherwise, if the tram receives the route created message from IXL and the traffic light is set to GO the tram can continue to follow its route. The activity TL fires and the OG4 output gate function is executed. The OG4 output function updates the place Topology with the tram index in the array position that indicates the section with Track Circuit and increases the place Location by one token. The predicate input of the IG3 gate models the fact that a tram can only move to the next line segment if it is not currently occupied by another tram, and only if the tram has received from IXL the connection request confirmation message and the route created confirmation message.

After the occurrence of one of the two situations, the tram passes the traffic lights and occupies the virtual Track Circuits that make the route unavailable for other requests from other trams, modeled with the place TC. The time it takes for the tram to cross the Track Circuits is represented by the LeaveTC activity, whose firing executes the code of the output function of the OG5 output gate (similar to OG2). If the tram is along an ordinary segment, put the token in the place PO. If the tram is in a Connection Request Tag segment, it inserts the token in the place ReqCon and updates the array of the extended place ReqCons in which the trams that forwarded the connection request to IXL are kept. Then, the Topology extended place is updated, with the unique index of the tram that forwarded the connection request, which is then in the i-th position of the segment with Connection Request Tag. It also inserts a token in the place Location. If the PO and Location places do not have tokens, and if the Topology extended place is at position 0, the OG5 output function inserts a token in the place Out and updates the FinishTime extended place array which is an array used to trace the end of trip times. Moreover, the firing of the LeaveTC activity determines the addition of a token in the place Out, which represents the exit of the tram from the route.

The upper-right part of the model represents the disconnection of the OBCU of the tram from IXL after the tram came out of the line segment with the Track Circuits. The predicate input of the IG5 gate enables the Confirm activity if at least one token is present in the place Out. The Confirm activity is deterministic with the RateIXL value and represents the time to send the disconnection ack message from OBCU to IXL.

C. The IXL SAN model

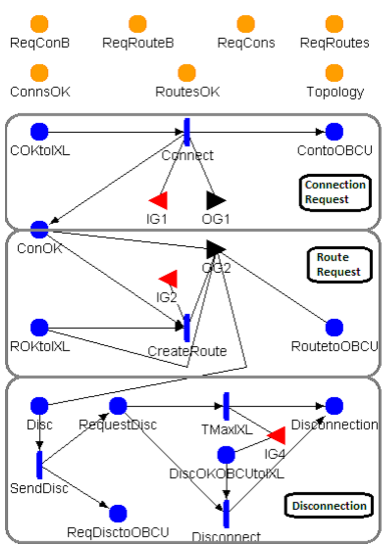

Fig. 3. IXL SAN model.

The model of Figure 3 represents the model of the IXL component concerning the management of connection and route requests received from the OBCU and the sending of acks and disconnection from IXL, after the appropriate checks. The model will be presented by discussing its sub-parts: Connection Request; Route Request; and Disconnection.

The connection request received from the Tram is modeled in the Connection Request box by the place COKToIXL. The input predicate of the IG1 input gate verifies the condition that the request for connection by a tram has arrived (with a certain index). The firing of the Connect activity executes the output gate OG1 function. This function guarantees the connection to the tram with the index that requested it, updating the extended place ReqConB, which is a buffer to manage requests in FIFO order. Furthermore, the Connect activity firing puts a token in the place ConToOBCU, which represents the connection between the IXL and the OBCU of the tram.

The management of the route request is modeled in the Route Request box. The place ConOK models the connection established with IXL. The place ROKtoIXL models the route request received from the OBCU. The input predicate of the IG2 input gate verifies if the connection between IXL and OBCU has been established with the tram index that requested it. The CreateRoute activity firing executes the output function of the OG2 output gate. The OG2 function checks if the route required by the tram is free via a function that checks if the Virtual Track Circuits are free from other trams. If the requested route is free, the function updates the array of the extended place RoutesOK with the index of the tram to which it is assigning the route and update the ReqRouteB buffer. Moreover, the function puts a token in the place RouteToOBCU that models sending the created route confirmation message to the OBCU and places a token in the place Disc, which models sending the IXL disconnect request to OBCU. Otherwise, if the route is not available, it returns a token to place ConOK and ROKtoIXL.

In the Disconnection section, the tram disconnection procedure is modeled. The presence of a token in the place Disc models the disconnection request sent by IXL to OBCU. The SendDisc transition has a deterministic distribution. The SendDisc firing determines the addition of a token in the ReqDisc place, which represents the disconnection request, and a token in the place ReqDiscToOBCU that models the sending of a disconnected request to OBCU. At this point, two things can happen. The TMaxIXL deterministic activity represents the time out with which the IXL frees the route even without having received the disconnection confirmation message from OBCU (IG4 input function). Otherwise, the Disconnect activity firing adds a token in the place Disconnection that models the disconnection between IXL and OBCU.

D. The Network SAN model

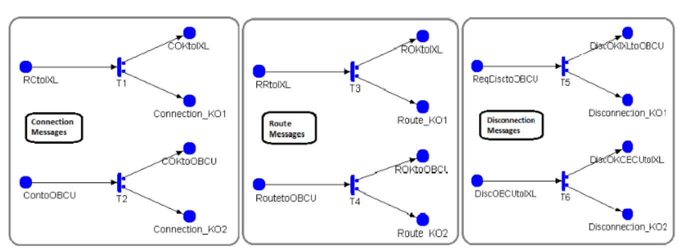

Fig. 4. Network SAN model.

The Network atomic model represents the communication channel between OBCU and IXL. The communication consists of the transmission of connection request, route request, and disconnection request messages, with respective replies. The atomic model relating to Network is shown in Figure 4.

The connection request message transmission from the OBCU to the IXL and the reply message are modeled in the Connection Messages box. The two activities T1 and T2 have a deterministic distribution with a value called RateNetwork. Both activities have two cases, which represent the probability that the transmission of the message may fail (with probability p) and the probability that the transmission will be successful (with probability 1-p). The behavior of the other boxes is similar to Route Messages and Disconnection Messages, respectively.

V. Analyzed Scenario and Analysis results

A. Definition of the Topology of the Line and fixed parameters

The considered topology was derived from an real-life existing line, and it is 16.3Km long, with 12 junction areas. Each junction area has a total length of 350 meters as follows: the distance from the Request Connection tag to Route Request tag is 50 meters; the distance from the Route Request tag to the Traffic Light is equal to 200 meters, and; the distance from the Traffic Light to the end of the track circuit segment is 100 meters. The first junction was inserted after 500 meters of free line and then the following distances between junctions were considered, which correspond to the lengths of the ordinary segments that separate two consecutive junction areas: 300 meters, 150 meters, 850 meters, 3850 meters, 250 meters, 650 meters, 1350 meters, 250 meters, 700 meters, 1700 meters, and 1550 meters. The topology has been modeled through the definition of the array global variable:

$$ShortRulesar[] = \{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 3, 0, 0, 0, 0, 0, 0, 1, 2, 3, 0, 0, 0, 1, 2, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 3, 0, 0, 0, 0, 0, 1, 2, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 3, 0, 0, 0, 0, 0, 1, 2, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 3\};$$

where: rulesar [i] = 0: the i-th segment of the line represents an ordinary segment (50 meters long); rulesar [i] = 1: the i-th segment of the line represents the segment where the tram detects the first virtual tag connection request; rulesar [i] = 2: the i-th segment of the line represents the segment where the tram detects the second virtual tag route request; rulesar [i] = 3: the i-th segment of the line represents the segment corresponding to the virtual track circuit that determines the occupation or not of the route.

The setting of the main parameters of the models, which are used for all the analyses presented in this section, representing a real-life setting is shown in Table I. In the rest of this section, the results of the analysis are presented considering the following system aspects: accuracy of the positioning system; time-out value for the activation of manual procedures; quality of the communication network; mobility of trams.

The two targeted metrics have been computed as it follows.

- The average Trip Time of each tram involved in the scenario has been computed using the C++ instruction $LastActionTime$ available in the Möbius simulator. This instruction has access to the current simulation clock, and returns the simulation time at which the last activity fired. The metric has been computed as difference between the simulation time at which a tram reaches the end of the line, and the simulation time at which the same tram started its trip.
- The average number of times the driver had to call OCC and activate the manual procedure has been computed through the definition of an instant of time rate reward variable that returns the number of tokens accumulated in place nOCC of the Tram model.

The results were obtained with the simulator provided by the Möbius tool, with a minimum of 10,000 batches, a relative confidence interval of 0.1 and a confidence level of 0.95. Therefore, the simulator stopped when at least 95% of the results were within the range of $\pm 10\%$ of the average value. The metrics of interest were defined as described in Section IV. The model includes different distributions such as uniform, exponential, and deterministic, for which it does not exist an analytic solver.

B. Analysis based on positioning accuracy

The first series of assessments concerned the analysis of the impact of positioning accuracy, which allowed us to define how much this quantity affects traffic management. The objective of this analysis is to understand the impact of accuracy (with $epsilon = 1, 3, 5, 10$ meters) in a scenario that we define as nominal, i.e., a scenario in which the communication network works perfectly ($p = 0$), the variability of the tram course is very limited ($Bound = 1\%$), and the time-out for the activation of the manual procedure is set to its typical value ($Tmax = 8$ seconds). A small bound could correspond to a line with a path separated from the normal traffic of cars/pedestrians, and therefore with times of crossing of the various segments of the line almost constant.

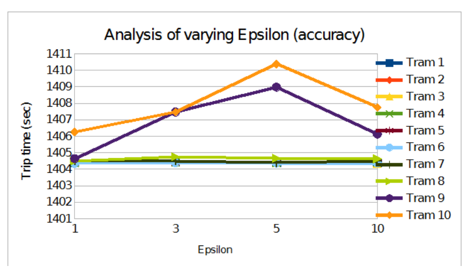

Fig. 5. Trip time depending on the Epsilon parameter (accuracy).

Figure 5 shows the results of the analysis. The 10 trams that run along the line are numbered in ascending order. Tram 1 is the first tram starting while Tram 10 is the last tram starting. The results show that in nominal conditions each tram takes around 1404 seconds (23.4 minutes) to cover the route in question. The last 2 trams accumulate a negligible delay, of 6 seconds in the worst case. Therefore, we can conclude that under nominal conditions the accuracy parameter has no impact on the performance of the tramway line considered. This is also the case for $epsilon = 0$, that is, the scenario equivalent to physical sensors. The average CPU time for the four experiments in this scenario was 7.868 seconds, and the maximum was 9.215 seconds. As a comparison, for 50 trams, the average CPU time for the four experiments in this scenario was 74.325 seconds, and the maximum was 76.971 seconds.

C. Analysis of varying time-out for the manual procedure activation

A second series of assessments concerned the analysis of the impact on the entire traffic management of the $TMax$ parameter (with $Tmax = 4, 8, 12, 16, 20$ seconds), which represents the maximum time a tram can wait in front of a traffic light placed at a standstill before contacting the OCC and then activating the manual procedure. The objective of this analysis is to understand the impact of the time-out in a nominal scenario , i.e., a scenario in which the communication network works perfectly ($p = 0$), the variability of the tram trend is very limited ($Bound = 1\%$), and the accuracy is set to its typical value ($Epsilon = 3$ meters). The results are presented in Figure 6. Also, in this case, it is possible to conclude that the $Tmax$ parameter, under nominal conditions, has no impact on the performance of the tramway line considered, since the delays accumulated in the worst case are of the order of a few seconds.

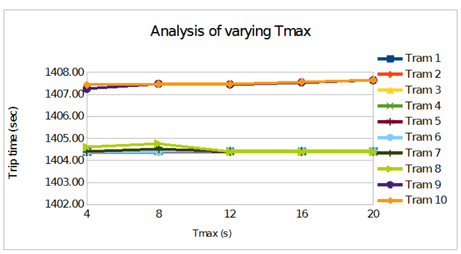

Fig. 6. Trip time depending on the TMax parameter.

D. Analysis at varying of the quality of the communication network

The third series of evaluations concerned the analysis of the impact of the parameter p (with $p = 0, 0.001$, 0.003, 0.005, 0.01) which made it possible to understand how much the probability that a message does not arrive at its destination affects the traffic management. The lack of communication between the architectural components of the system causes delays in the journey. For example, if the response message from IXL for the request route message had not been delivered to the OBCU, the tram should still contact the OCC even in the event of a traffic light set to GO. The results presented in Figure 7 show that the impact of network quality on the performance of the tramway line considered may be significant.

With $p = 0.001$ (or if 1 message is lost every 1000 transmitted between OBCU and IXL) an average delay is

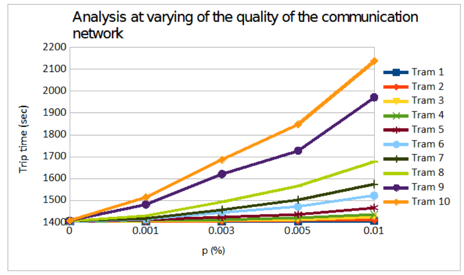

Fig. 7. Trip time when varying the quality of the communication network.

already obtained on all trams equal to about 17 seconds with respect to the nominal travel time, with maximum peaks of 111 seconds of delay for the last tram number 10 (about 8% delay compared to the nominal travel time of 1404 seconds). With $p = 0.005$ (or if 5 messages are lost per 1000 transmitted) an average delay is already obtained on all trams of around 64 seconds, with peaks of 443 seconds of delay for the last tram (equal to more than 30% of delay compared to the nominal travel time). The results with $p = 0.01$ are significantly the worst, with an average delay of the last tram (the number 10) equal to more than 50% of the nominal travel time.

On the same scenario, the number of trams that must activate the manual procedure with the OCC was also analyzed. At each activation by a tram, there will be a double effect. The tram will accumulate a delay defined by the Delay parameter (120 seconds), and this delay can be propagated as a cascading effect to the following trams. Thus, this delay influences the tramway traffic of the entire area. The results are presented in Figure 8. As expected, the accumulation of delays in the travel time shown in Figure 7 is proportional to the increase in the number of calls to the OCC, and therefore to the activation of the manual procedure.

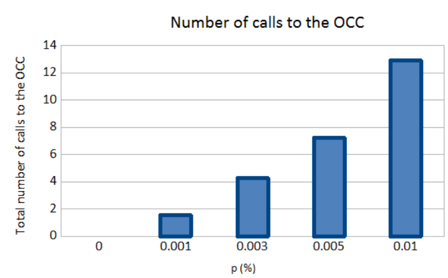

Fig. 8. Total number of calls to the OCC as the network quality changes.

E. Analysis with different tramway mobility characteristics

In this section, we analyze the impact of the value of the Bound parameter which represents the impact of the physical characteristics of the tram line and the level of interaction between the tram and the rest of the city traffic. The mobility of a tram can be more or less affected by city traffic. For example, a line completely separate from the normal traffic of cars and pedestrians will have less variable trip times (therefore with a smaller bound) with respect to a scenario in which the tram line is shared with cars, with pedestrian crossings, etc. (and therefore with a larger bound). The time taken by the tram to travel an ordinary segment of the line was modeled with uniform distribution: $LowerBound : (ol/AverageVelocity) * (1 - Bound)$; $UpperBound : (ol/AverageVelocity) * (1 + Bound)$.

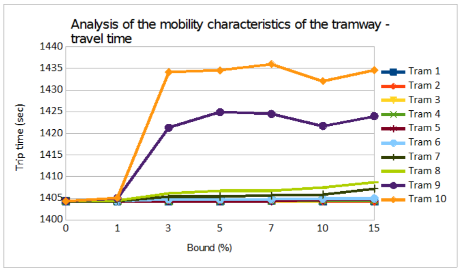

Fig. 9. Trip time as the Bound varies.

Increasing the Bound value will increase the variability of the trip time of each tram. The Bound is an indication of the variability of the tram course. A small Bound could correspond to a line separated from the normal traffic of cars and pedestrians, and therefore with crossing times of the various segments of the line almost constant. As the Bound increases, the variability of the time taken by each tram trip increases. The results are shown in Figure 9. As the variability of tram movement increases, trip times increase slightly to reach a different upper limit for each tram. The worst case is that related to Tram 10 (the last tram that enters the scenario), whose average trip time increases by about 32 seconds compared to the average trip time of 1404 seconds. Therefore, we can conclude that the bound has a rather negligible impact compared to the average trip time of the trams.

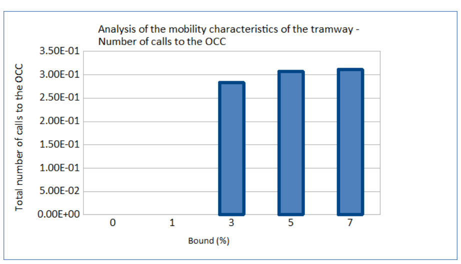

Fig. 10. Total number of calls to the OCC as the Bound varies.

Moreover, the number of trams that have to activate the manual procedure with the OCC has also been analyzed. We can observe a delay in the management of the tram traffic of that area. The results are presented in Figure 10. The results are consistent with that of Figure 9 since the accumulated delays increase proportionally with the increase in the number of calls to the OCC (activations of the manual procedure).

F. Combination of previous analyzes

In the previous sections, we analyzed the impact of one parameter at a time concerning the metric of interest. In this section, we consider two different parameter settings.

Parameter	Best value	Worst value
TMax	20 seconds	4 seconds
p	0	0.01
Bound	1%	7%
Epsilon	1 meter	10 meters

TABLE II
PARAMETERS FOR THE ANALYSIS IN THE BEST AND WORST CASES.

The best condition is obtained by setting the parameters to minimize the delay accumulated by the trams. The worst condition is obtained by setting the parameters to maximize the accumulated delay by the trams. Table II summarizes the setting of the parameters for the best and worst case analysis.

The results are presented in Figure 11. Moving from the best to the worst case, one can see a general increase in average trip times, which becomes more significant for trams from number 5 (average delay of about 1 minute) to number 10 (average delay of almost 15 minutes). The average travel time on all trams goes from 1404 seconds in the best case to an average of 1639 seconds in the worst case, which corresponds to an average delay of about 4 minutes per tram.

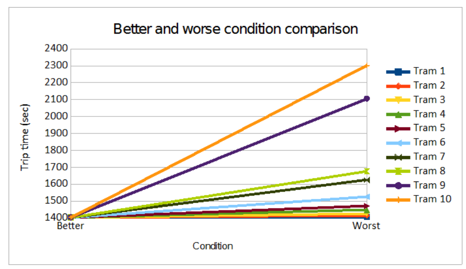

Fig. 11. Trip time in the worst and best case.

It can also be noted that the combination of the epsilon, TMax, p, and Bound parameters in the worst case determines a delay in the tram trip time which is greater than the sum of the individual contributions extracted respectively from Figure 5, Figure 6, Figure 7 and Figure 9. This emerging behavior is due to the interdependence between the same parameters. For example, summing the individual contributions of epsilon, TMax, p, and Bound for Tram 10 as calculated in the previous analysis, an overall delay of 771 seconds is obtained (3 seconds of delay for $epsilon = 10$ meters, 3 seconds for $TMax = 4$ seconds, 733 seconds for $p = 0.01$, 32 seconds for $Bound = 7\%$), while from Figure 11 we can see how the overall delay in the worst case is equal to $2300 - 1404 = 896$ seconds, i.e., more than 2 minutes in addition to the delay calculated as a sum of individual contributions.

G. Analysis at varying of the time-out for the activation of the manual procedure - with the failure of a tram

In this section, the analysis of the trip time at varying of the Tmax parameter has been combined with the presence of a temporary failure of a tram on the line. We simulated the presence of a failure in Tram 1 (the first tram) starting from time 700 seconds, whose effect is to stop the tram, and its repair after another 1800 seconds (30 minutes for the repair of the failure). The results are presented in Figure 12. Tram

1, that is affected by the failure, undergoes the greatest delay and it propagates to all other trains. It is interesting to note that the parameter $Tmax = 20$ seconds, in this scenario, is much more powerful than all other configurations since the delay accumulated by all the trams, following the first one, is significantly reduced.

The delay of Tram 10, for example, is reduced from 2873 seconds to just 1826 seconds. This improvement is because $Tmax = 20$ seconds allows the following trams absorb the cascade delays caused by Tram 1, allowing the trams to remain a few more seconds in front of the traffic lights set to STOP without having to activate the manual procedure. Therefore, it can be concluded that in conditions of a highly congested tram network (as in the case induced by the temporary tram failure) the time-out parameter for the OCC call can have a significant impact on network performance and in particular on the ability to reabsorb accumulated delays and decongest the network.

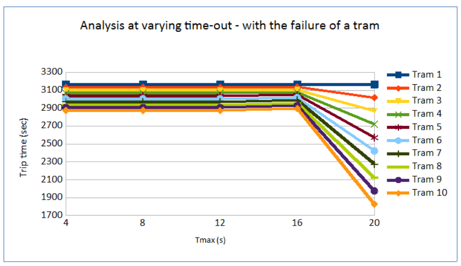

Fig. 12. Trip time with the failure of a tram.

VI. Concluding remarks

In this paper, we presented the quantitative analysis of the SISTER system, applying stochastic modeling techniques. The general objective of the quantitative analyses presented in this work is to analyze the impact of the new architectural solutions and its key parameters on the traffic that can be supported by a tramway. The main advantages of this new solution are the low cost and easy re-configuration, compared to the traditional system. A model based on Stochastic Activity Networks was developed, which was then solved using the simulator integrated into the Möbius tool.

From the analysis of the results, related to a portion of the existing tramway line, it was possible to conclude that in nominal traffic conditions (no problem of loss of messages, no tram affected by failure) the accuracy and time-out parameters for the OCC call have no impact on the performance of the tramway line considered. The parameter of bound that characterizes the variability in the mobility of trams along the route has a negligible impact compared to the average travel time of the trams. In conditions of high congestion of the line, for example, due to a temporary failure of a tram, the time-out parameter for the OCC call can have a significant impact on network performance and in particular on the ability to reabsorb the accumulated delays and therefore the ability to decongest the line. Moreover, the quality of the communication network between OBCU and IXL has a very significant impact on network performance.

The content of this work is also a solid starting point for further extending the analysis scenarios. Moreover, thanks to the modularity and generality of the adopted modeling methodology, it will be possible to extend the models to analyze a more complex tram network, for example, composed of several routes and with intersections of lines, and to extend the models in the case of introduction of new functionalities of the system. As future work, we also envision a detailed comparison with $epsilon = 0$ (traditional system) for all the scenarios, a scalability analysis, and more outage scenarios.

Acknowledgment

This work has been partially supported by the REGIONE TOSCANA POR FESR 2014-2020 SISTER SIgnaling & Sensing Technologies in Railway application, and by the European Unions Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 823788.

References

[1] L. Pugi, F. Grasso, and G. Rossi, “Energy simulation of tramway systems, simplified and efficient models,” in *2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I CPS Europe)*, June 2018, pp. 1–6.

[2] H. Nagai and S. Kurahashi, “Agent-based urban dynamics simulation to analyze effects of street attractiveness and tramway for an urban sprawl,” in *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, vol. 2, July 2017, pp. 660–665.

[3] O. Dib, M.-A. Manier, L. Moalic, and A. Caminada, “A multimodal transport network model and efficient algorithms for building advanced traveler information systems,” *Transportation Research Procedia*, vol. 22, pp. 134 – 143, 2017, 19th EURO Working Group on Transportation Meeting, EWGT2016, 5-7 September 2016, Istanbul, Turkey.

[4] D. Sierra, J. Briceo, H. Buitrago, B. Rozo, L. Montecchi, and I. Mura, “Probabilistic modeling of failure domino effects in chemical plants,” in *2018 Eighth Latin-American Symposium on Dependable Computing (LADC)*, Oct 2018, pp. 57–66.

[5] S. Chiaradonna, F. Di Giandomenico, and G. Masetti, “Analyzing the impact of failures in the electric power distribution grid,” in *2016 Seventh Latin-American Symposium on Dependable Computing (LADC)*, Oct 2016, pp. 99–108.

[6] E. Borba and E. Tavares, “Stochastic modeling for performance and availability evaluation of hybrid storage systems,” *Journal of Systems and Software*, vol. 134, pp. 1 – 11, 2017.

[7] F. Corman and P. Kecman, “Stochastic prediction of train delays in real-time using bayesian networks,” *Transportation Research Part C: Emerging Technologies*, vol. 95, pp. 599 – 615, 2018.

[8] M. Biagi, L. Carnevali, M. Paolieri, and E. Vicario, “Performability evaluation of the ertms/etcs level 3,” *Transportation Research Part C: Emerging Technologies*, vol. 82, pp. 314 – 336, 2017.

[9] W. H. Sanders and J. F. Meyer, *Stochastic Activity Networks: Formal Definitions and Concepts*. Berlin, Heidelberg: Springer, 2001, vol. 2090, pp. 315–343.

[10] W. H. Sanders, “Integrated frameworks for multi-level and multi-formalism modeling,” in *Proceedings 8th International Workshop on Petri Nets and Performance Models (Cat. No.PR00331)*, Sep. 1999, pp. 2–9.

[11] W. H. Sanders, T. Courtney, D. Deavours, D. Daly, S. Derisavi, and V. Lam, “Multi-formalism and multi-solution-method modeling frameworks: The mbius approach,” in *Proceedings of the Symposium on Performance Evaluation - Stories and Perspectives*, Dec. 2003, pp. 241–256.