

RESEARCH

Open Access



# A stochastic modeling approach for traffic analysis of a tramway system with virtual tags and local positioning

Leandro Dias da Silva<sup>1\*</sup> , Paolo Lollini<sup>3,4</sup>, Diamantea Mongelli<sup>2</sup>, Andrea Bondavalli<sup>3,4</sup> and Gianluca Mandò<sup>5</sup>

\*Correspondence:

[leandrodias@ic.ufal.br](mailto:leandrodias@ic.ufal.br)

<sup>1</sup>Computing Institute, Federal University of Alagoas, Maceió, Brazil  
Full list of author information is available at the end of the article

## Abstract

Traditional 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, like the communications with the ground infrastructure and the interlocking system. This approach is not easily scalable and maintainable, and it is costly. The SISTER project designed new architectural solutions for addressing the previous problems based on the virtualization of the sensors and on the local positioning of each tram. The key 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 traffic that can be supported by the tramway network. This paper presents a stochastic modeling approach for analysing the performability of a tramway system based on the SISTER architectural solutions, aiming to identify the parts of the tramway network that are more critical and sensible to the variation of the traffic conditions and to the setting of the key architectural parameters. We build a model using Stochastic Activity Networks and run sensitivity analyses on (i) the accuracy of the positioning, (ii) the different SISTER parameters, and (iii) considering possible outages temporarily blocking the journey of a tram. This analysis allows to properly set and fine-tune the key architectural parameters, to understand the impact of the accuracy on the positioning, to understand the impact of the outages.

**Keywords:** Performability analysis, Stochastic activity networks, Tramway, Interlocking system

## Introduction

The SISTER project (REGIONE TOSCANA POR FESR 2014-2020 SISTER “Signaling & Sensing Technologies in Railway application”<sup>1</sup>) 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.

<sup>1</sup><http://www.progetto-sister.com/>

The innovative architectural solutions developed in SISTER are mainly based on the virtualization of physical sensors and on the local positioning system of each tram. Tram-interlocking communications are no longer activated at the achievement of Tram Detector (Tag) as in traditional tram systems but are activated when the local position calculated by each tram corresponds to a virtual connection request or route request tag. 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.

The general objective of the quantitative analyses presented in this work is to analyze and understand the impact of these new architectural solutions and its key parameters on the traffic that can be supported by a tram network. 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 analyses presented in this paper are performed by developing a stochastic model based on stochastic activity networks (SAN) [1] that can be simulated using the simulator integrated in the Möbius tool [2, 3]. Models have been developed in a parametric and modular way, so to facilitate their extension and their application to different scenarios and different topologies of the tramway networks.

This paper is an extended version of [4], where we introduced the basic elements of the modeling approach targeting a quite simple scenario with one single route and one single track only. In this work, we provide the following key extensions:

- We extend the SAN models to allow the modeling and analyses of more complex and realistic scenarios with multiple routes, intersections, and shared routes.
- We provide new data structures to allow the definition of new topologies of tramway networks in a parametric way.
- We defined and included a new metric to analyze the distribution of the traffic towards the whole network, allowing to identify the parts of the tramway networks that are more critical and sensible to the variation of the environment conditions and to the setting of the key architectural parameters.
- Finally, we provide a more detailed explanation of the overall SISTER system functioning, of the interactions among its constituent components, and the new functions and data structures used in the extended model to deal with multiple routes and intersections.

The paper is organized as follows. “[SISTER main architectural components](#)” section outlines the key SISTER architectural elements. “[Modeling the operational scenario](#)” section 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 “[Stochastic activity network \(SAN\) models](#)” section, while results are presented and

discussed in “[Analyzed scenario and analysis results](#)” section. Related works are discussed in “[Related work](#)” section, while “[Concluding remarks](#)” section concludes the paper.

**SISTER main architectural components**

This section describes the aspects of the SISTER sub-systems that have been captured by the models.

The SISTER project builds on the idea of developing a solution for rail and tram transport through the development, integration, and validation of innovative virtual signaling technologies. Currently, the tram detection is carried out through track circuits and axis counting system.

An abstract view of the SISTER architecture is shown in Fig. 1, and it includes the Location Determination System (LDS), the Operational Control Center (OCC), the On-Board Control Unit (OBCU), and the Interlocking (IXL). SISTER implements new tram detection technologies, such as Local Determination System (LDS) based on Global Navigation Satellite System (GNSS) and inertial technologies. The GNSS is based on EGNOS (European Geostationary Navigation Overlay Service) that is a Satellite Based Augmentation System (SBAS). It consists of a network of about 40 ground stations and 3 geostationary satellites. Ground stations determine accuracy data of the satellite navigation systems and transfer it to the geostationary satellites. The LDS component also includes an Inertial Measurement Unit (IMU), which represents the Measurement Unit, and a Data Fusion algorithm that calculates the position of the tram.

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 Virtual Tags are placed on the computer on board of the tram on a pre-loaded map, which identifies the points of the track where the connection and route requests’ messages should be send

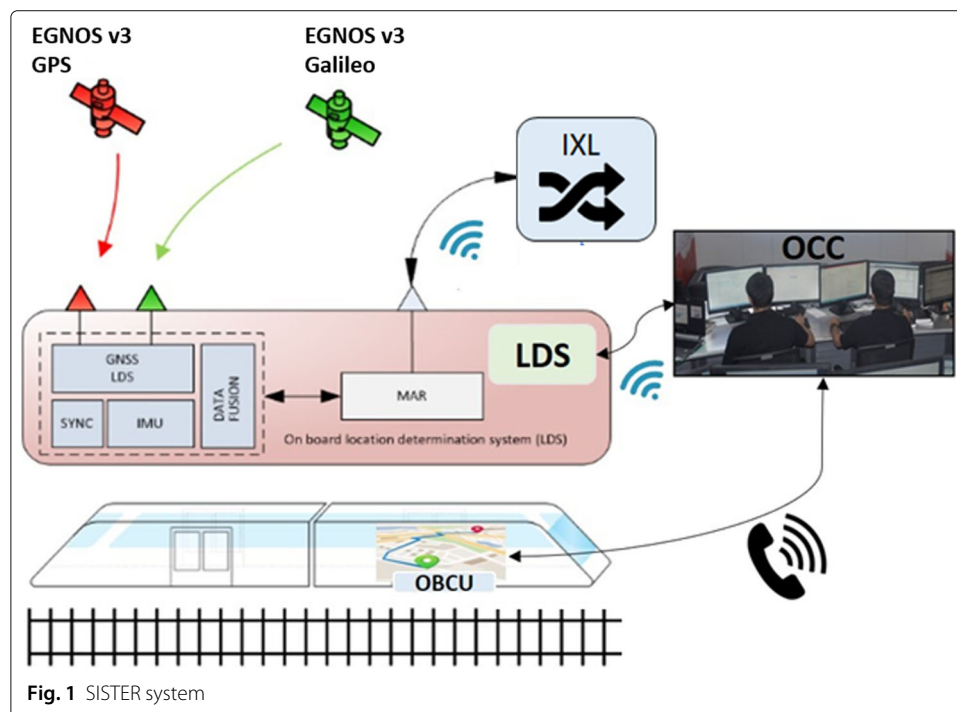


Fig. 1 SISTER system

to the interlocking system (IXL). 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 architectural components of the new system, their functionalities, and the related assumptions necessary for modeling are described in the following.

#### **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) integrates the Traffic Management software application, which implements the Automatic Vehicles Location function. This component receives location information from the trams and places them on a synoptic graphical interface which allows the traffic operator to manage the movement of the trams along the track lines. 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 tram signal set to STOP, after a specific time-out the driver must contact OCC for understanding what is the problem and for activating a manual procedure.

The OCC provides, with established procedures, information to the driver on how to proceed in certain situations of tram stopping in front of the tram signal. The OCC can also carry out communications on the tram for general information to passengers. It is the driver's responsibility to follow the manual procedures since it is a visual guide.

#### **On-Board Control Unit (OBCU)**

The On-Board Control Unit (OBCU) component 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 IXL. Subsequently, when it detects the second virtual tag (route request) on the map, it sends the route creation request message to IXL. If IXL, after appropriate checks, sends the route created message, then the tram signal 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 Track Circuit corresponds to a SIL4 tram sensor placed in some critical places of the lines, typically in correspondence with a junction area, and it is used to limit the access to a part of a track. 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.

The OBCU receives location data from LDS once per second and sends it to IXL also once per second.

### **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 the receipt of the route request from OBCU of the tram, the IXL performs the feasibility check to evaluate if the requested route is available (not occupied). If it is available then it commands the maneuvering boxes, sets the tram 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.

Some necessary assumptions for the correct development of the IXL model had to be defined. The interlocking bandwidth must be wide at least to contain the requests of the maximum number of trams that it can manage simultaneously on a specific area. The connection network is always available between the various components along the route, regardless of its correct functioning or not.

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 *completely* exited from the last virtual Track Circuit, which means that it will be freed Accuracy meters after the tram position communicated by LDS.

IXL uses the position of the tram, considering the maximum error Accuracy when going to occupy the track circuit. IXL frees the track circuit only when the tram has completely exited from the last virtual track circuit, considering the Accuracy.

### **Focus: interactions between the SISTER architectural components while approaching a junction area**

Initially, the tram is outside the junction area, it is not connected to any interlocking, it proceeds at a certain speed, and it is connected to the OCC. The tram periodically performs the following actions:

- Detects its virtual position  $L_v$  calculated using the Sensor Fusion Algorithm embedded in the LDS component and then:
- It connects at regular intervals to the OCC, sends its virtual position  $L_v$  and its own identifier, then disconnects from the OCC, and compares its virtual position  $L_v$  with the onboard map for the detection of Virtual Tags.
- The OBCU detects the virtual tag indicating the presence of a junction area. This virtual tag also contains information for connecting to the interlocking that manages the junction area.

- The OBCU of the tram, therefore, detects the virtual route request tag.
- The OBCU sends a connection request to the interlocking along with its own identifier.
- The interlocking receives the connection request, analyzes it, and sends a message to the OBCU to confirm that the connection has been established.
- When OBCU detects the connection request Virtual Tag on the map, it sends to IXL a message with the request and its own identifier. IXL accepts the request.
- The OBCU sends the request for a specific route together with its identifier to the interlocking.
- The interlocking receives the route request and performs the feasibility check of the requested route based on the status of the monitored virtual track circuits, using as input the position of all the trams it is connected to. If the requested route is free from other trams it creates the route, commands all exchanging in the correct position, sets the tram signal to GO, and sends a route created confirmation message to OBCU.
- The tram enters the junction area overcoming the tram signal and periodically sends its position Lv and its identifier to the interlocking it is connected to.
- The interlocking periodically calculates the Location Referencing function, which establishes the occupation of the track circuits and verifies that the sequence of occupation of the track circuit by the tram is that foreseen by the assigned route.
- As soon as the interlocking detects the occupation of the first track circuit in the junction area, this tram signal is set as STOP.
- The tram then enters the junction area and sends its position once per second. IXL monitors the progress of the tram on the route with a Location Referencing function and sets the tram signal as STOP.
- The interlocking clears the route assigned to the tram when the tram exits the virtual tag corresponding to the route release condition.
- When the interlocking detects the exit of the tram from the last track circuit of the junction area, it sends a disconnection request to the tram and then it disconnects.
- OBCU receives the request for disconnection from the interlocking and sends a confirmation response.
- Upon receipt of the confirmation response from the OBCU of the tram, the interlocking disconnects the communication with the OBCU of the tram.

### **Operational scenario and objectives of the analysis**

In this section, we introduce the reference operational scenario targeted by the modeling approach, and the definition of the metrics of interest.

The operational scenario that we analyze is derived from a real Chinese tramway network and consists six tram lines (routes) with a series of junctions and intersections. The junctions are near the platforms of the line, where the tram stops for passengers to get on or off. The scenario, schematized in Fig. 2, consists of a certain number of trams that intend to cover the part of the considered route, following the departure times as defined in the planning phase of the route timetable. The route is characterized by the presence of virtual connection request (RC) and route request (RR) tags, tram signals for platform access, and the corresponding Track Circuit (TC). A TC in correspondence of a tram platform is used, for example, when the platform is positioned after a blind curve that

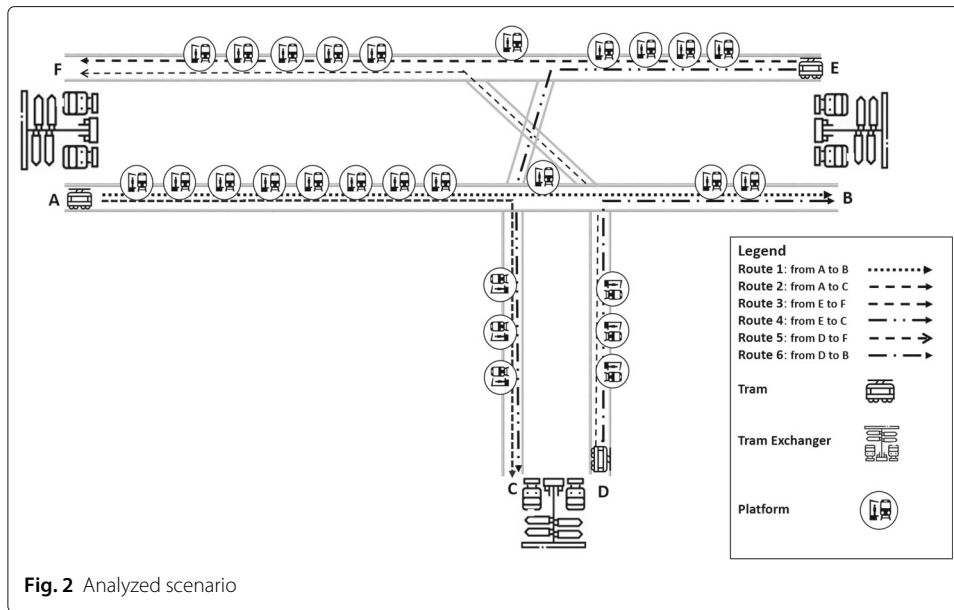


Fig. 2 Analyzed scenario

prevents an approaching tram from detecting the presence of another tram occupying the platform.

The routes are discretized as the following. For all the routes, the junction area has a total length of 180 meters as follows: the distance from the Request Connection tag to Route Request tag is 50 m; the distance from the Request Route tag to the tram signal is equal to 85 m, and the distance from the tram signal to the end of the track circuit segment is 45 m.

- Route 1.** The trams on Route 1 start from terminal A, on the left side of Fig. 2, and share the first part of the track with trams on Route 2 (8 platforms). The trams on Route 1 arrive at terminal B on the right side of Fig. 2. Route 1 is about 15.3 km long and has 11 junction areas. The first junction is located after 500 m from the beginning of the line. Then, the following distances between junctions were considered: 150 m, 1400 m, 250 m, 1500 m, 200 m, 3500 m, 100 m, 1000 m, 2150 m, and 2550 m.
- Route 2.** The trams on Route 2 also start from terminal A, but arrive at terminal C, on the bottom of Fig. 2. Trams on Route 2 share the last part of the track with trams that come from Route 4 (3 platforms). Route 2 is about 10 km long, with 11 junction areas. The first junction is located after 500 m from the beginning of the line and then the following distances between junctions were considered: 150 m, 1400 m, 250 m, 1500 m, 200 m, 3500 m, 100 m, 250 m, 50 m, and 50 m.
- Route 3.** The trams on Route 3 start from terminal E, right side of Fig. 2, and share the first part of track with the trams on Route 4 (4 platforms). The trams on Route 3 arrive at terminal F on the right side of Fig. 2. Route 3 is about 11.2 km long and has 10 junction areas. The first junction is located after 500 m from the beginning of the line. Then, the following distances between

junctions were considered: 200 m, 1900 m, 1300 m, 100 m, 350 m, 3100 m, 100 m, 1600 m, and 150 m.

- *Route 4.* The trams on Route 4 also start from terminal E, and arrive at terminal C, on the bottom of Fig. 2, where they share this part of the track with trams that come from Route 2 (3 platforms).

Route 4 is about 5.6 km long and has 7 junction areas. The first junction is located after 500 m from the beginning of the line. Then, the following distances between junctions were considered: 200 m, 1900 m, 1300 m, 250 m, 50 m, and 50 m.

- *Route 5.* The trams on Route 5 start from terminal D, on the bottom of Fig. 2, and proceed to the terminal F, on the left side of Fig. 2. The trams on Route 5 share the beginning of the track with trams on Route 6 (3 platforms), and they share the last part of the track with trams on Route 3 (5 platforms).

Route 5 is about 7.5 km long and has 8 junction areas. The first junction is located after 500 meters from the beginning of the line. Then, the following distances between junctions were considered: 150 m, 100 m, 350 m, 3100 m, 150 m, 1600 m, and 150 m.

- *Route 6.* The trams on Route 6 start from terminal D, on the bottom of Fig. 2, and proceed to the terminal E, on the right side of Fig. 2. The trams on Route 6 share the beginning of the track with trams on Route 5 (3 Platforms), and share the last part of the track with trams on Route 1 (2 platforms).

Route 6 is about 6.8 km long and has 5 junction areas. The first junction is located after 500 m from the beginning of the line. Then, the following distances between junctions were considered: 150 m, 500 m, 2150 m, and 2550 m.

### **Objectives of the analysis and metrics of interest**

The analysis aims to quantitatively evaluate the impact of some key system parameters on tramway traffic of a real scenario. The main metrics of interest related to the performance of the tram network are:

- *Average time required by each tram to cross the considered route:* calculated as the difference between the time the tram reaches the end of the route and its departure time.
- *Average number of activation of the manual procedure:* In the event that the driver has to wait in front of a tram signal set to STOP for a time exceeding a predefined time-out, the driver must contact the ground control unit (OCC) to activate the manual procedure and know how to proceed. The activation of the manual procedure by the driver will result in a delay in the entire management of the tram traffic in that area, and this metric intends to estimate the average number of the manual procedure activation (sight guide).
- *Average occupancy time of each segment of a route, i.e.,* the average time a segment of a route is occupied by a tram. This metric allows to check which are the parts of the route where trams spend more time, thus emphasizing the critical parts of the tramway network.

The main objective of the analysis is to evaluate the impact of some key parameters of the SISTER system on these metrics, so as to be able to highlight any critical issues depending on their configuration. In particular, we intend to study the previous indicators as the following system parameters change:



- *Accuracy of the onboard positioning system*: it is the maximum error, expressed in meters, of the positioning data communicated by the Local Determination System (LDS) to the other elements of the architecture (OnBoard Control Unit- OBCU and Interlocking-IXL). It is intended to analyze whether and under what conditions this parameter can have an impact on the management of tram signals, on route requests from other trams and therefore on the overall tramway traffic.
- *Time-out value for the activation of manual procedures (call to the OCC)*: this is the maximum time that each tram can wait in front of a tram signal set to STOP, and after which it must contact the ground operations center (OCC) to activate the manual procedure and know how to proceed, accumulating a delay in its path. It is intended to evaluate whether and under what conditions this parameter can have an impact on the tramway performance.
- *Quality of the OBCU-IXL communication network*: the quality of the communication between the architectural components of the system causes delays in the journey through the route. For example, if the response message from IXL for the tram route request had not been delivered to the OBCU, the tram should still contact the OCC even in the event of a tram signal set to go ahead.
- *Level of interference between the mobility of the trams and the surrounding environment*: the mobility of a tram can be more or less influenced by normal city traffic. For example, a line with a path completely separate from normal car/pedestrian traffic will have less variable travel times compared to a scenario in which the tram line is shared with cars, with pedestrian crossings, etc. This type of analysis is aimed at understanding the impact of the variability of tram movement on the performance of the tram network.
- *Temporary malfunction of a tram*: the temporary malfunction (outage) of a single tram, for example, due to an accidental failure, can determine the congestion of part of the tramway. We intend to analyze the tramway capability to reabsorb any delays caused by unexpected and external events to the system.

### **Modeling the operational scenario**

In this section, we start bridging the gap between the system and how it has been modeled. Specifically, we discuss the key modeling aspects of the SISTER sub-systems that have been captured by stochastic models, referring to the operational scenario depicted in Fig. 2. In “[Stochastic activity network \(SAN\) models](#)” section, we will provide the detailed description of the corresponding SAN models and how they can be composed to represent the behavior of the whole system.

All the model parameters introduced in this section are also reported in Table 1.

### **Tram**

The main aspects captured by the Tram model are the topology of the tram network (including virtual tags, tram signals, and track circuit); the movement of a single tram on the route; the accuracy of the positioning system (LDS component); the driver’s call to the OCC; and the exchange of messages with IXL (OBCU component), i.e., the connection request and route request messages. This is the model that represents the typical operational scenario of a single tram, which runs through a portion of the line crossing the virtual tags, tram signals, and track circuits as described in the previous section. The LDS

**Table 1** A real-life setting of model parameters

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: 1%, 3%, 5%, 7%, 10%, and 15%.
Epsilon	The maximum error on the tram position calculated by the LDS system (Accuracy)	variable: 1, 3, 5, and 10.
TMax	The maximum time that a tram can wait in front of a tram signal before contacting the OCC	variable: 4, 8, 12, 16, and 20.
p	Probability that a message does not arrive at its destination	variable: 0, 0.001, 0.003, 0.005, and 0.01.
av	The starting time interval between the trams	variable: 20 and 30 s.
AverageVelocity	Average tram speed	14.0 m/s
RateIXL	IXL processing time in response to connection request and route request messages	0.1 s
RateNetwork	Time of transmission of a message from OBCU to IXL or from IXL to OBCU	0.08 s
Delay	The additional delay accumulated by a tram following the activation of the manual procedure	120 s
lentc	The length of Track Circuit	45 m
nl	The total number of segments	569
nt	The total number of trams	36
ol	The length of the ordinary segment	50 m
rrtotl	The distance between the Route Request Tag and the tram signal	85 m
pt	The time at the tram platform	20 s

component of the architecture of the new system has not been explicitly captured in the model, but we modeled its effects in terms of the Accuracy of the tram position estimated by the component.

Each route 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 tram signal 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 tram signal, 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 terminal at

the time established in 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 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 tram signal set as STOP, after a time defined by the  $TMax$  system parameter, the driver 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 if not occupied. When IXL detects that the segment with the track circuit is no more occupied by 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.

### 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 tram signals. 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 of the virtual Track Circuits with respect to the route that was requested. If available, it sets the tram signal to GO, manages the switching boxes, and sends the route created message to OBCU. There is just one parameter for this model,  $RateIXL$ , that corresponds to the average response time to each request.

### Network

The network model represents the communication channel for the exchange of messages between tram (OBCU) and interlocking (IXL), i.e., it is the channel used for the transmission of the following messages:

- Connection request messages from OBCU to IXL
- Confirmation messages for connection creation from IXL to OBCU
- Route request messages from OBCU to IXL
- Confirmation route creation messages from IXL to OBCU

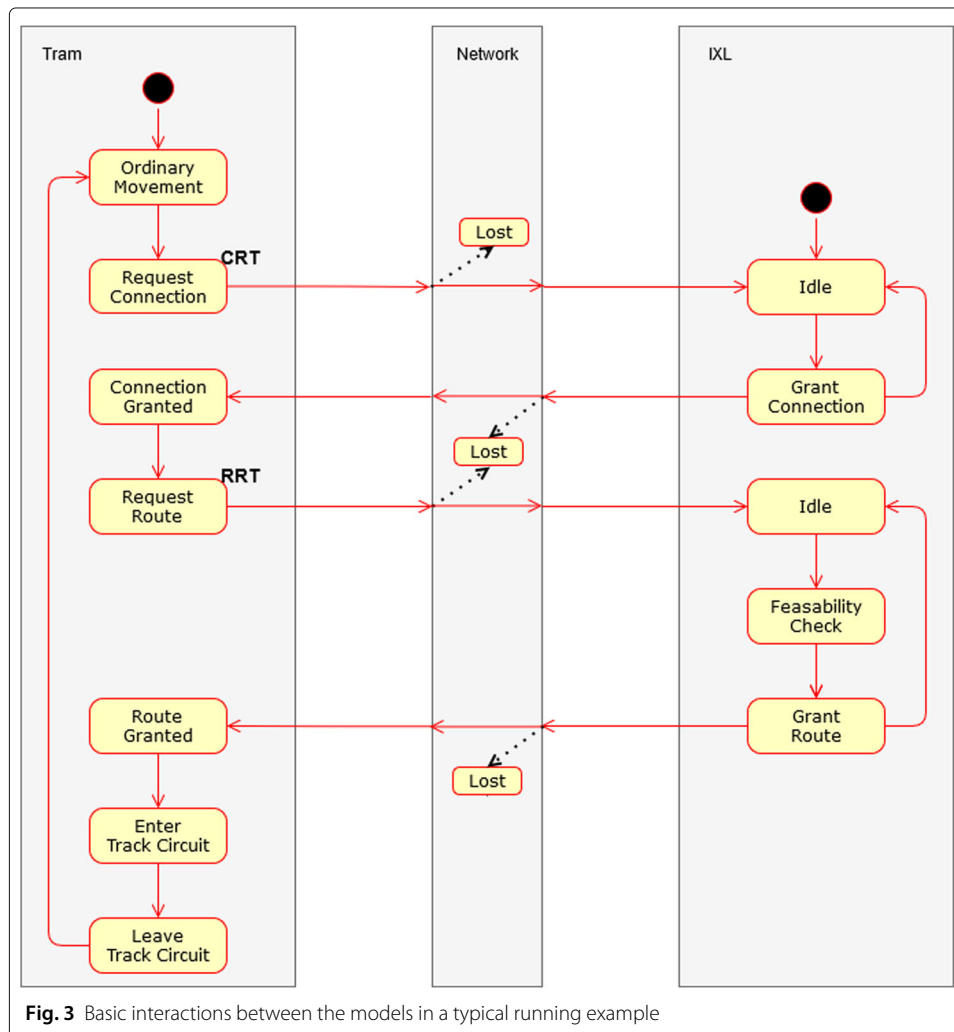
The quality of the communication channel is captured through a parameter  $p$  that represents the probability that a single message is not delivered to the recipient, which allow us to study the impact of the quality of communication on the performance of the tramway.

The tram model, which characterizes the movement of a single tram in the network, is therefore replicated as many times as the number of trams in the scenario. The replicated model of tram is then composed with the IXL model and with the network model to form the overall model of the system. Further details on the replication and composition of the models are provided in the “Stochastic activity network (SAN) models” section.

**Basic interactions between the models**

For the sake of clarity, in Fig. 3, we depict the basic interactions between the three models (tram, network, and IXL) in a typical running example where a tram is approaching to a free junction area and the network is properly working.

The network model represents the interface between the tram and the IXL models, since all the messages exchanged between the tram and the interlocking system pass through this model. The interactions between the tram and the IXL models are always



triggered by the tram model in correspondence of the two virtual tags: at the Connection Request Tag (CRT) the tram sends a request connection message to IXL, which grants the connection and replies with the connection granted message; then, when the tram reaches the Route Request Tag (RRT) it sends a route request message to IXL that will run the feasibility check and grant the requested route. Therefore, from the modeling point of view, IXL implements two control loops: one for granting the connection and the other for performing the feasibility check and granting the route.

### **Stochastic activity network (SAN) models**

In this section, we provide the technical details of the stochastic activity network model built for the analysis of the tram network considering the specificities of the SISTER system. The full technical description is available at the repository<sup>2</sup>.

The model was developed using the stochastic activity networks (SAN) [1] modeling formalism and the Möbius [2, 3] tool. SAN formalism has been selected for its modeling features that allow to properly represent the behavior of the target system, among which the possibility to represent both timed and instantaneous activities, to associate generally distributed random variables to activity time distribution functions, to define case probabilities for modeling uncertainty associated with the completion of an activity, and to build the system model as composition of different submodels. Then Möbius has been selected as the only available mature tool for editing and solving SAN models, which integrates both analytical solvers and a discrete event simulator.

The key aspects captured by the models and the main parameters are those already described in the previous sections. In this section, we provide the technical details of the model, discussing their actual implementation.

### **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 (i) different sequences of junction area and (ii) different numbers of routes with intersections, which (iii) may share different parts of a track. The objective has been achieved through a parametric definition of the data structures used in the model and sharing some extended places between the models. Further details will be provided during the description of the models.

---

<sup>2</sup><https://drive.google.com/open?id=1KneJXELmXkMRYLZUzU9oFsjefTfpi2y2>

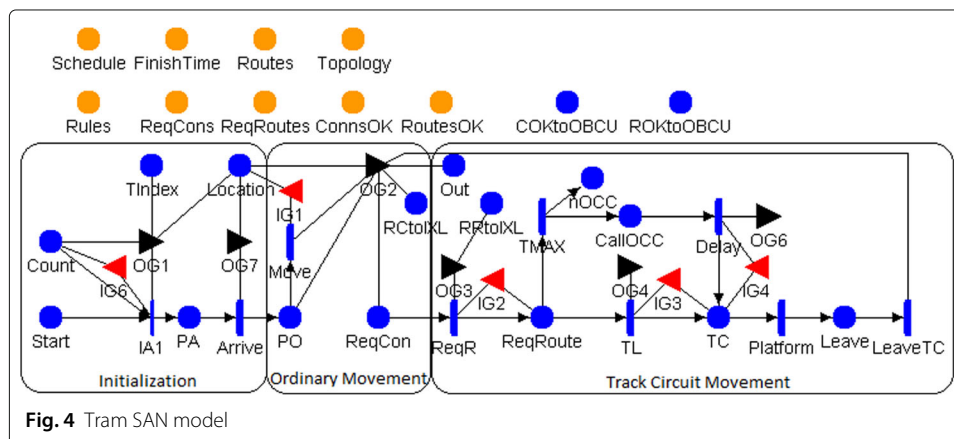
**The tram SAN model**

The tram atomic model is shown in Fig. 4. The model will be presented by discussing its sub-parts. The initialization part is responsible to set up the whole scenario, that is, the data structures for the routes and topology. The Ordinary Movement part is responsible for the movement of the trams. This is the general part of the model. The Track Circuit Movement part is responsible for the interaction with the IXL model for route creation to occupy the track circuit. Most of the logic and rules concerning the tram movement are coded inside the input and output gates, within the input and output functions. In the following, we will explain parts of the codes in the gates and relate them with the movement of the trams.

**Initialization**

The part of the model connected to the output gate OG1 is used to make non-anonymous replicas of the Tram model and to initialize the model with the tram 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. 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, and place Count contains *nt* tokens. 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  $TIndex \rightarrow Mark() = nt - Count \rightarrow Mark() - 1$ , where *nt* is the total number of trams and Count place is 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*), where the *i*-th element indicates the position of each tram along the line. Initially, all the trams are in the first position of its specific route in the array. Moreover, all the trams are waiting to enter the scenario according to their scheduled timetable. When activity Arrive fires, the position of the tram is updated as the tram moves along the route. The place Location is set according to the starting position of each tram, by output gate OG7 as  $Location \rightarrow Mark() = startar[TIndex \rightarrow Mark()]$ , where *startar* is the array with all starting positions for each route and tram. The IG6



**Fig. 4** Tram SAN model

input gate plays the fundamental role of initializing the parameters of the model and in particular of defining the topology of the scenario and the departure times of the trams.

The specification of the scenario is done in the IG6 input gate by the definition of a set of arrays (global variables) that represent the roles of the various segments of the line, the routes, the trams, and the schedule. A specific setting of the arrays is explained in the analysed scenario in “[Analyzed scenario and analysis results](#)” section. 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 routes are defined in the *routesar* array global variable. It is an  $nr \times nl$  matrix, where *nr* is the number of routes in the scenario and *nl* is the number of segments in the scenario. The definition of each route is the sequence of segments the tram should move on to cover that specific route. The segments that are not part of the route of a tram are filled with the last index of the array, indicating that these segments are not part of its route. The *routes* array, with size *nt*, defines the routes for each tram, i.e., it associates a route to each tram.

The tram scheduling is parametric and it 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 its route in the array. The variable *at* is a variable of type double which represents the time that elapses between the departure of two consecutive trams. The departure time of the *i*-th tram is  $Schedule \rightarrow Index(i) \rightarrow Mark()$ . The departure time of each tram is defined in the IG6 gate by the code in Listing 1, but it can be modified considering another scheduling. In this implementation, Tram *i* and Tram *i* + 1 are *at* seconds apart (we considered *at* = 30). As there are six routes, the departure time of a tram in the same route is 180 s (3 min).

**Listing 1** IG6 - output function

```
int i;
double t;
for (i=0, t=1.0; i < nt; i++, t=t+at){
    Schedule->Index(i)->Mark()=t;
}
}
```

The Arrive activity models the beginning of the *i*-th tram route along the scenario, following the scheduled time. The Arrive activity has a deterministic distribution with parameter  $Schedule \rightarrow Index(i) \rightarrow Mark() \rightarrow Mark()$ . The Arrive activity firing causes the insertion of a token in the place PO that represents the actual departure of the *i*-th tram.

As explained in this section, the model is completely parametric and can have any number of routes, of any size, any number of trams, covering any route in the scenario, with

individual or group schedules. This is done by setting the four arrays in the initialization part of the model *rulesar*, *routesar*, *routes*, *schedule*.

### Ordinary Movement

The Move activity represents the time spent by the tram to pass through an ordinary segment of the line (not corresponding to any virtual tag) and has a uniform distribution as can be seen in Table 1. The variable *ol* is a global 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 segment if it is not currently occupied by another tram, as shown in Listing 2.

**Listing 2** IG1 - predicate

```
Topology->Index ( Routes->
Index ( TIndex->Mark() ) ->
Index ( Location->Mark() ) -> Mark() ) ->
Mark() == 0
```

This code means that the next segment the tram is moving to should be free (marking equal to 0). To access the next segment in the topology array, the model uses the current location (*Location* -> *Mark()*) and tram individual identification (*TIndex* -> *Mark()*) as index to the *Routes* array. The location index in the routes array identifies the next segment the tram needs to go in that route.

When the Move activity fires, the token is removed from the place PO, and the OG2 gate output code shown in Listing 3 is executed.

**Listing 3** OG2 - output function

```
//Not terminating movement
if ( Location->Mark() < ( nl - 1 ) ) {
  if ( Rules->Index ( Routes->
    Index ( TIndex->Mark() ) ->
    Index ( Location->Mark() ) -> Mark() ) ->
    Mark() == 0 ) {
    PO->Mark ( ) = 1;
  } else {
    //Request Connection segment
    if ( Rules->Index ( Routes->
      Index ( TIndex->Mark() ) ->
      Index ( Location->Mark() ) -> Mark() ) ->
      Mark() == 1 ) {
      ReqCon->Mark ( ) = 1;
      RCtoIXL->Mark ( ) = 1;
      ReqCons->Index ( TIndex->Mark() ) ->
      Mark ( ) = 1;
    }
  }
  //Free the current location
  Topology->Index ( Location->Mark() ) ->
  Mark ( ) = 0;
```



```

//Update the location
Location->Mark() = Routes->
Index ( TIndex->Mark()) ->
Index ( Location->Mark()) ->Mark ();
//Update the topology
Topology->Index ( Location->Mark()) ->
Mark () = ( TIndex->Mark () + 1 );
} else {
//Terminating movement
//Remove token from movement
PO->Mark () = 0;
//free current location
Topology->Index ( Location->Mark()) ->
Mark () = 0;
Location->Mark () = 0;
Out->Mark () ++;
//save finish time
FinishTime->Index ( TIndex->Mark()) ->
Mark () = LastActionTime ;
}

```

The output function of the OG2 gate checks whether the tram is along an ordinary segment, and a token is put back 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. Then, the Topology extended place is updated, with the (unique) index of the tram that forwarded the connection request, which is then at the *i*-th position of the segment with Connection Request Tag. It also updates the marking in the place Location.

If place Location indicates the tram is in the last segment, a token is inserted in the place *Out* and the array of the extended place *FinishTime* is updated, which is the array used to keep track of the ending time of the journey of each tram, i.e., the time at which each tram completes its 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, that is the *i*-th tram finish time minus the starting time.

### **Track Circuit Movement**

The ReqR activity represents the crossing time of the segment with Virtual Request Route tag, distributed according to the uniform distribution: Lower Bound:  $((rrtotl - \epsilon)/AverageVelocity) * 0.99$ ; Upper bound:  $((rrtotl - \epsilon)/AverageVelocity) * 1.01$ . The variable *rrtotl* is a global variable that represents the length of the segment with Route Request Tag, while *epsilon* is the global 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 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, like the same test in IG1 explained before. The Output function in OG3 (see Listing 4) 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.

**Listing 4** OG3 - output function

```
//flags the route request
ReqRoutes->Index ( TIndex->Mark() ) - >
Mark () = 1;
//free current location
Topology->Index ( Location ->Mark() ) - >
Mark () = 0;
//update the location
Location ->Mark () = Routes ->
Index ( TIndex->Mark() ) - >
Index ( Location ->Mark() ) - > Mark ();
Topology->Index ( Location ->Mark() ) - >
Mark () = ( TIndex->Mark () + 1 );
//send request to IXL
RRtoIXL->Mark () ++;
```

Moreover, OG3 updates the place Topology with the tram index in the array position that indicates the segment with the Route Request Tag, and updates the Location place marking and increments the RRtoIXL marking 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 tram signal must be set to GO before it can be overtaken.

The input predicates of input gates IG3 and IG4 model the fact that a tram can only move to the next line segment if it is not currently occupied by another tram, as it is done in the input gates IG1 and IG2. Moreover, the predicate input of the IG3 gate checks also if the tram has received the connection request confirmation and the route created confirmation messages from IXL. If the tram receives the route created message from IXL and the tram signal is set to GO, the tram can continue to follow its route.

The TL (tram signal) activity represents the tram signal that is managed by IXL, together with the requests received from the tram and the shunting stations. The activity TL fires and the code in Listing 5 is executed in the OG4 output gate function.

**Listing 5** OG4 - output function

```
//free current location
Topology->Index ( Location ->Mark() ) - >
Mark () = 0;
//update location
Location ->Mark () = Routes ->
Index ( TIndex->Mark() ) - >
Index ( Location ->Mark() ) - > Mark ();
//update Topology
```

```
Topology->Index ( Location ->Mark() ) - >  
Mark () = ( TIndex->Mark () + 1 );  
//clear the connection request  
ConnsOK->Index ( TIndex->Mark() ) - >  
Mark () = 0;  
//clear the route request  
RoutesOK->Index ( TIndex->Mark() ) - >  
Mark () = 0;
```

The OG4 output function updates the place topology with the tram index in the array position that indicates the segment with Track Circuit and updates the place Location.

If the tram does not receive the confirmation message of the creation of the route from IXL or the tram signal is set to STOP, after a deterministic time indicated by the *tmax* variable in the TMAX activity, the tram driver must activate the manual procedure and call OCC. In this case, a token is inserted in the nOCC and CalOCC 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 updates the place Location (moves to the next segment), similar to what is done in the output gate OG4.

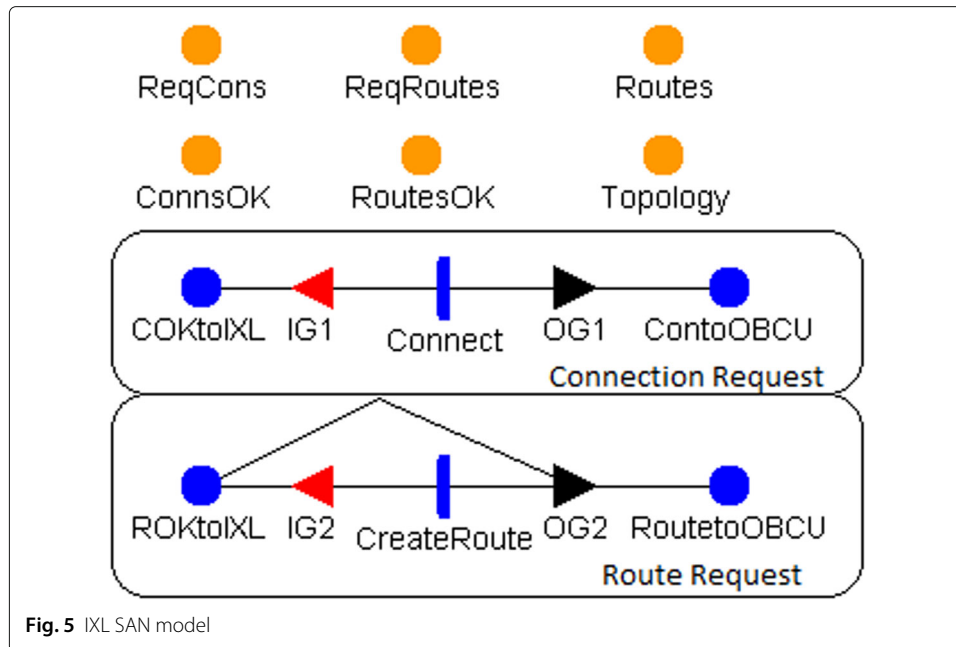
After the occurrence of one of the two situations described before (TL or TMax firing), the tram passes the tram signal and occupies the virtual Track Circuits that make the route unavailable for other requests from other trams, modeled with the place TC. After that, the tram stops at the platform for  $pt = 20$  s, modeled by the deterministic timed activity Platform. The time required by the tram to cross the Track Circuits is represented by the LeaveTC activity, whose firing executes the code of the output function of the OG2 output gate again, and the dynamics of the tram movements repeats until it reaches another connection request segment. If the tram is along an ordinary segment, the token is added in the place PO. If the tram is in a Connection Request Tag segment, it adds 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 at the *i*-th position of the segment with Connection Request Tag. It also updates the marking in the place Location.

### The IXL SAN model

The model of Fig. 5 represents the model of the IXL component concerning the management of connection and route requests received from the OBCU and the sending of acks, after the appropriate checks. The model will be presented by discussing its sub-parts: Connection Request and Route Request.

#### Connection Request

The connection request received from the tram model is modeled by the place COK-ToIXL. The input predicate of the IG1 input gate verifies the condition that the request for connection by a tram has arrived ( $COKtoIXL - > Mark() > 0$ ). The firing of the



Connect activity executes the output gate OG1 function, presented in Listing 6. This function guarantees the connection to the tram with the index that requested it, updating the extended place ConnsOK. 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.

**Listing 6** OG1 - output function

```

\\ for every tram
for (int i=0;i<nt;i++){
  \\ if it is requesting
  if (ReqCons->Index(i)->Mark()==1) {
    \\ grants the connection
    ConnsOK->Index(i)->Mark()=1;
    \\ clear the request
    ReqCons->Index(i)->Mark()=0;
    \\ acks the request
    ConToOBCU->Mark()++;
  }
}

```

### Route Request

The management of the route request is modeled in the Route Request part of the IXL model. The place ROKtoIXL models the route request received from the OBCU. The input predicate of the IG2 input gate verifies if there is a connection request between IXL and OBCU ( $ROKtoIXL \rightarrow Mark() > 0$ ).

When the CreateRoute activity fires, the output function of the OG2 output gate is executed (see Listing 7). 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 assigned the route. Moreover, the function puts a token in the place RouteToOBCU that models the sending the created route confirmation message to the OBCU. Otherwise, if the route is not available, it returns a token to place ROKtoIXL.

**Listing 7** OG2 - output function

```
// for every tram i
for (int i = 0; i < nt; i++) {
  // if tram i is requesting route
  if (ReqRoutes->Index(i)->Mark() == 1) {
    // for every segment in the topology
    for (int j = 0; j < (nl-1); j++) {
      // finds the tram
      if (Topology->Index(j)->Mark()==(i+1)){
        // if not end of the line
        if (Routes->Index(i)->Index(j)->Mark()
        <(nl-1)){
          // if the track circuit ahead is free
          if (Topology->Index(Routes->Index(i)
          ->Index(j)->Mark()->Mark() == 0) {
            // creates the route
            RoutesOK->Index(i)->Mark()=1;
            // clear the request
            ReqRoutes->Index(i)->Mark()=0;
            // acks the route
            RoutetoOBCU->Mark()++;
          } else { // if not free
            ROKtoIXL->Mark()++; // try again
          }
        } else { //if end of the line
          // creates the route
          RoutesOK->Index(i)->Mark()=1;
          // clear the request
          ReqRoutes->Index(i)->Mark()=0;
          // acks the route
          RoutetoOBCU->Mark()++;
        }
      }
      break;
    }
  }
}
```

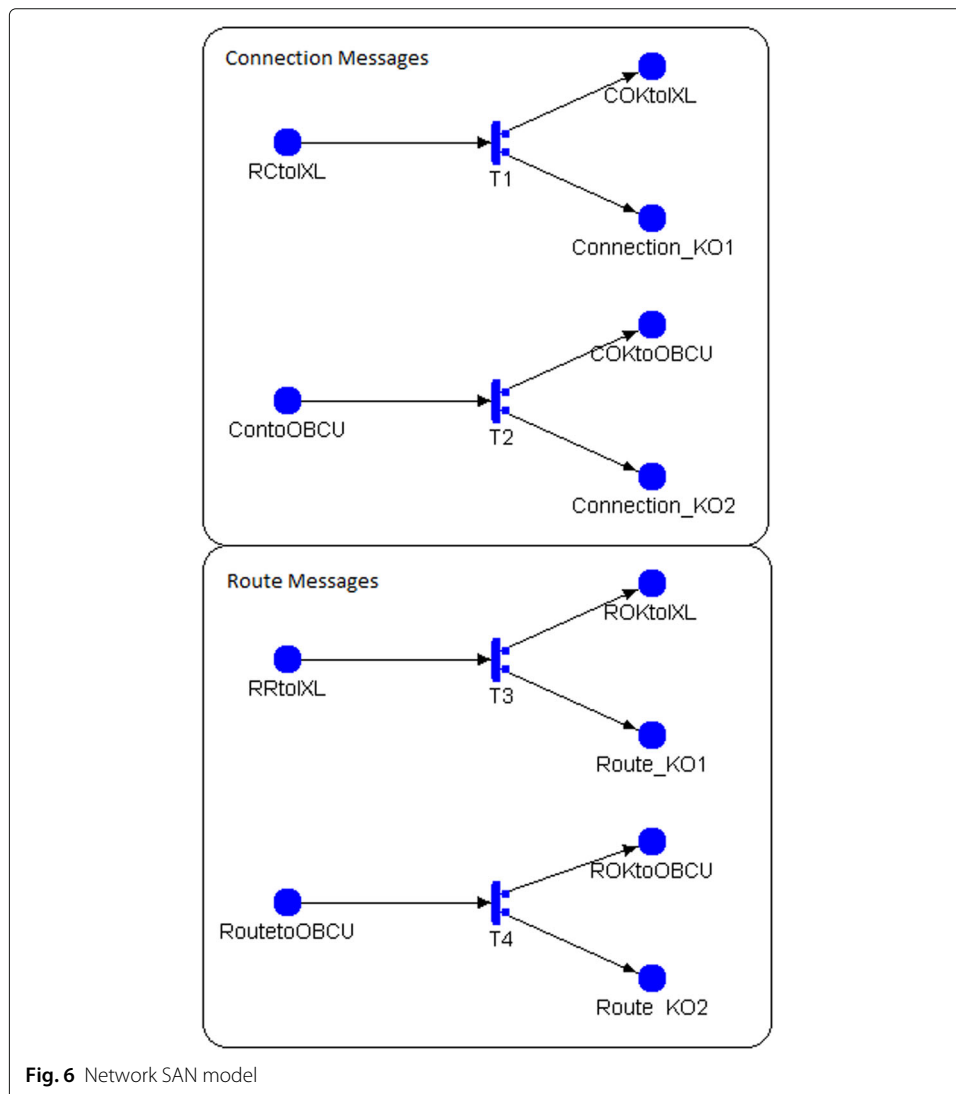
**The network SAN model**

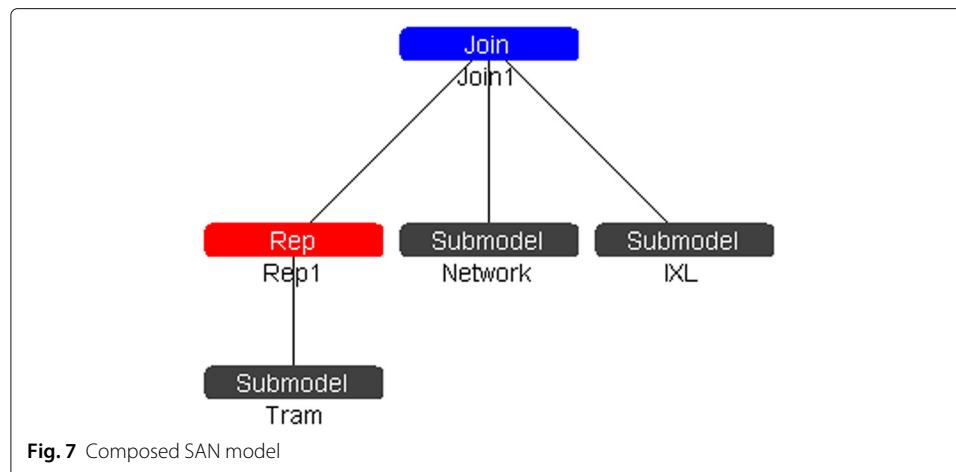
The network atomic model represents the communication channel between OBCU and IXL. The communication consists of the transmission of connection request and route request, with respective replies. The atomic model relating to Network is shown in Fig. 6.

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 Route Messages box is similar to the Connection Messages box.

**The composed SAN model**

The overall model of the system is shown in Fig. 7. The Tram model, which characterizes the running of a single tram in the scenario, has been replicated  $nt$  times using the REP





composition operator. The replication can be used to represent a certain number of trams running along the line.

Tram models replicated in this way share the following places which are common to all replicas: Count, Out, nOCC, RCtoIXL, RRtoIXL, COKtoOBCU, ConnsOK, ROKtoOBCU, Schedule, FinishTime, Routes, Topology, Rules, ReqCons, ReqRoutes, ConnsOK, and RoutesOK.

The replicated tram model is then composed with the IXL model and with the network model using the composition operator JOIN. The composition operation composes the models sharing all the places with the same name, thus forming the overall model of the system under examination. Therefore, the places RCtoIXL, RRtoIXL, COKtoOBCU, and ROKtoOBCU are shared between tram and network models. The places COKtoIXL, ROKtoIXL, ContoOBCU, and RoutetoOBCU are shared between IXL and network models. Finally, the extended places ReqCons, ReqRoutes, Routes, ConnsOK, RoutesOK, and Topology are shared between Tram and IXL models.

### Analyzed scenario and analysis results

The analyzed scenario consists of six routes with a series of junctions near the platforms, where the tram stops for passengers to get on or off. We considered six trams per routes, 36 in total, following the departure times as defined in the planning phase of the line timetable. The typical schedule for the analyzed scenario in this work is departure times of 30 s between trams, that is  $at = 30$ . As there are six routes, the departure time of a tram in the same route is 180 s (3 min).

### Definition of the topology of the line and fixed parameters

The considered topology was derived from a real-life existing tramway network, as defined in “Operational scenario and objectives of the analysis” section. The discretization process of the distance between junctions is based on ordinary segments  $ol$  which correspond to the lengths of the ordinary segments that separate two consecutive junction areas. In this work, we use  $ol = 50$ , that is, 50 m. The topology has been modeled through the definition of the *rulesar* array global variable. The total length of the *rulesar* is the interleaving of all the routes, defined in the global variable  $nl$ , that for this analyzed

scenario is  $nl = 569$ . As an example, the first 500 m and first junction are defined as illustrated in Listing 8.

**Listing 8** Array of rules definition

```
short rulesar [569] = {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 m 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.

The model is parametric, which means we can have any number of segments, junctions, intersections, routes, and trams. To define the number of segments, we use the above-mentioned  $nl$  global variable, in this work  $nl = 569$ . The global variable  $nt$  is used to define the number of trams (in this work  $nt = 36$ ). The junctions and intersections are defined by a triple 1,2,3 in the *rulesar* as exemplified before. To define the routes, we use a matrix global variable *routesar*. As an example, the beginning of the Route 1 is defined as shown in Listing 9. Each element in this matrix represents a unique identifier of a track segment (track-segment ID).

**Listing 9** Array of routes definition

```
short routesar [6][569] = {{1, 2, 3, 4, 5, 6, 7, 8,
9, 10, 11, 12, 13, ...}};
```

For each route, we have an entry in the *routesar* matrix. The numbers in a route represent the next segment in the route of the trams. For each movement the model looks at the *rulesar* array to see the rule of the segment. If it is an ordinary movement the model just checks if the segment is free and the tram moves ahead. If it is a rule 1 segment it triggers the connection request procedure with IXL. If it is a rule 2 segment it triggers the route request procedure with IXL. If it is a rule 3 segment, it waits until the route is created for the acknowledge message from IXL. As we have the interleaving of routes, not all the segments are used by every route in the *routesar*. The number in the array just points to next segment in the route, not necessarily the next one in the array. When it is the last segment of a route, it just points to the last segment, that in this case is 568.

To define which route a tram runs, we define in the input gate IG6 of the initialization part of the SAN tram model a route array to initiate each tram to its specific route, as shown in Listing 10.

**Listing 10** Tram to routes association definition

```
short routes [] = {0, 1, 2, 3, 4, 5, ...};
```

That means the first tram, position 0 in the array, will follow route 1, position 0 in the *routesar* matrix, and it follows like that for every tram. In this case, for 36 trams, 6 in Route 1, 6 in Route 2, etc.

Finally, we have to set the starting point of each tram. This is done by another array as shown in Listing 11.



**Listing 11** Starting positions definition

```
short startar [] = {0, 0, 299, 299, 531, 531, ...}
```

That is, the first and second tram will start in position 0 of the *routesar*, trams three and four in position 299 and trams five and six in position 531. And it follows like that for every tram.

The setting of the main parameters of the models is shown in Table 1, representing a real-life setting. 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, and mobility of trams.

The three targeted metrics have been computed as 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:  $return((Tram \rightarrow FinishTime - Index(i) \rightarrow Mark() - Tram \rightarrow Schedule \rightarrow Index(i) \rightarrow Mark()) / nt)$ ;
- 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:  $return((1.0 * Tram \rightarrow nOCC \rightarrow Mark()) / nt)$ ;
- The average occupancy time metric for a given segment in a track has been computed as the average time a segment of a route is occupied by a tram. For each segment of a route, we defined an interval of time rate reward variable that accumulates a reward 1 for every unit of time the specific segment is occupied by a tram, 0 otherwise, e.g., for segment ID = 20 the rate reward is defined as:  $return(Tram \rightarrow Topology \rightarrow Index(20) \rightarrow Mark())$ ;

The results were obtained with the simulator provided by the Möbius tool, with at least 100 batches and 1000 batches maximum, and with 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 model includes different distributions such as uniform, exponential, and deterministic, for which it does not exist an analytic solver.

The simulations have been performed on a Intel Core i7-4500U CPU @1.8GHz, with 8 GB of RAM, and each experiment converged in 4034 s on average.

For space reasons, we focused on the analysis of Route 4 only, which shares the last part of the track with Route 2 and results to be the most critical route and the most sensitive to parameters changes.

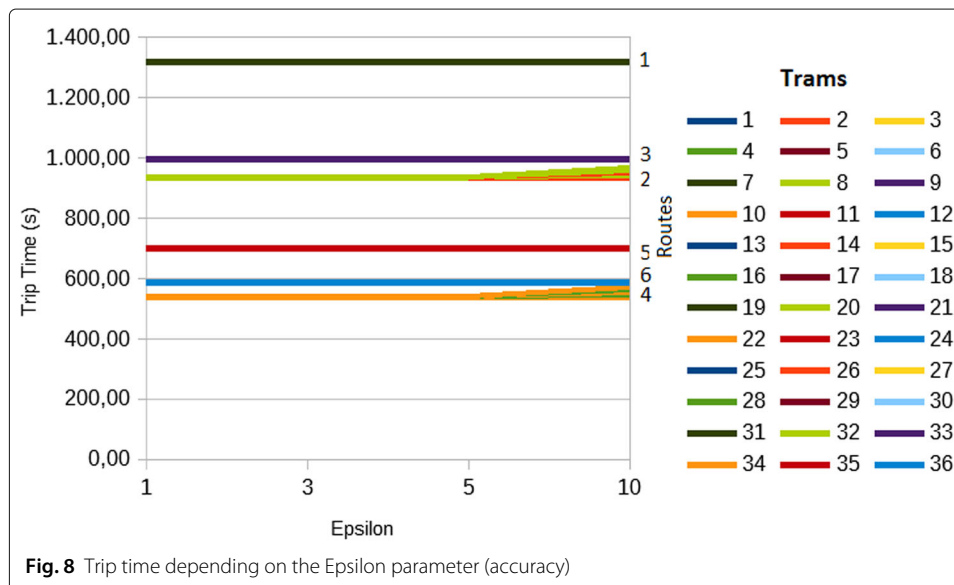
**Analysis based on positioning accuracy**

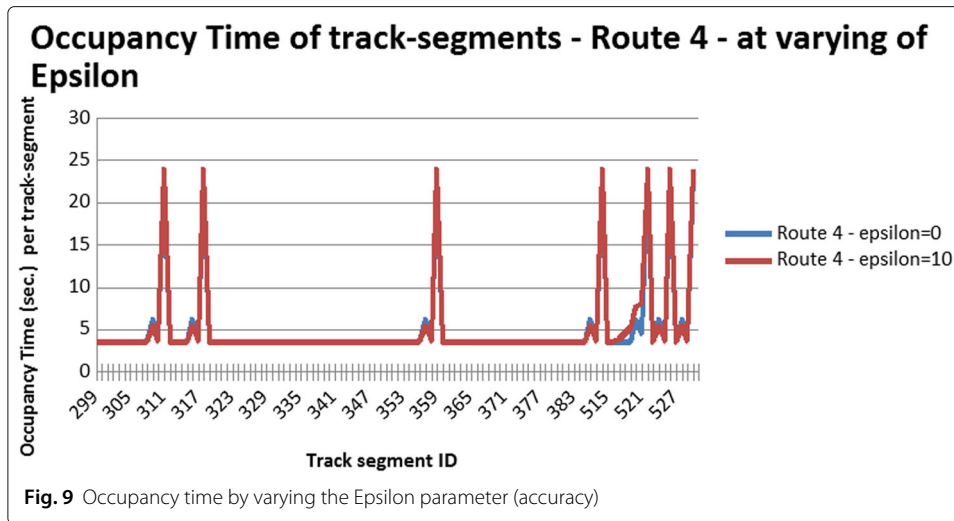
The first series of assessments concerned the analysis of the impact of positioning accuracy, which allowed us to define how much this quantity affects traffic management. The objective of this analysis is to understand the impact of accuracy (with *epsilon* = 1, 3, 5, 10

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 ( $tmax = 8$  s). 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.

Figure 8 shows the results of the analysis. The 36 trams are numbered in ascending order. Tram 1 is the first tram starting while Tram 36 is the last tram starting. Trams 1, 7, 13, 19, 25, 31 belong to Route 1, and Trams 2, 8, 14, 20, 26, 32 belong to Route 2, and so on. The results show that in nominal conditions a tram takes around 1315 s (21.92 min) to cover Route 1, 933 s (15.55 min) to cover Route 2, 997 s (16.62 min) to cover Route 3, 537 s (8.95 min) to cover Route 4, 699 s (11.65 min) to cover Route 5, and 586 s (9.77 min) to cover Route 6. The last trams on Routes 2 and 4 accumulated a delay of 33 and 30 s, respectively, in the worst case for  $epsilon = 10$ . Therefore, we can conclude that under nominal conditions the accuracy parameter has no impact on the performance of the analyzed tramway network. This is also the case for  $epsilon = 0$ , that is, the scenario equivalent to physical sensors.

The accuracy has a very negligible impact on the traffic distribution in all the analyzed routes. In Fig. 9, it is shown the traffic load for Route 4, for  $epsilon = 0$  and  $epsilon = 10$ . The values correspond to the average occupancy time of each segment composing the route. The segments are identified with a unique track-segment ID, that for Route 4 goes from 299 to 388, and from 515 to 530. The highest peaks (7 peaks in total - 24 s each) correspond to the Track-Circuit tags (e.g., at track-segment ID=311), the smaller ones (7 peaks in total - 6 s each) to the Request Route tags where the trams are waiting in front of the traffic-light (e.g., at track-segment ID=309). We can note that the occupancy time in the route is almost insensible to epsilon variations. The sum of the occupancy time values of all the segments in the route corresponds to the trip time computed in Fig. 8.

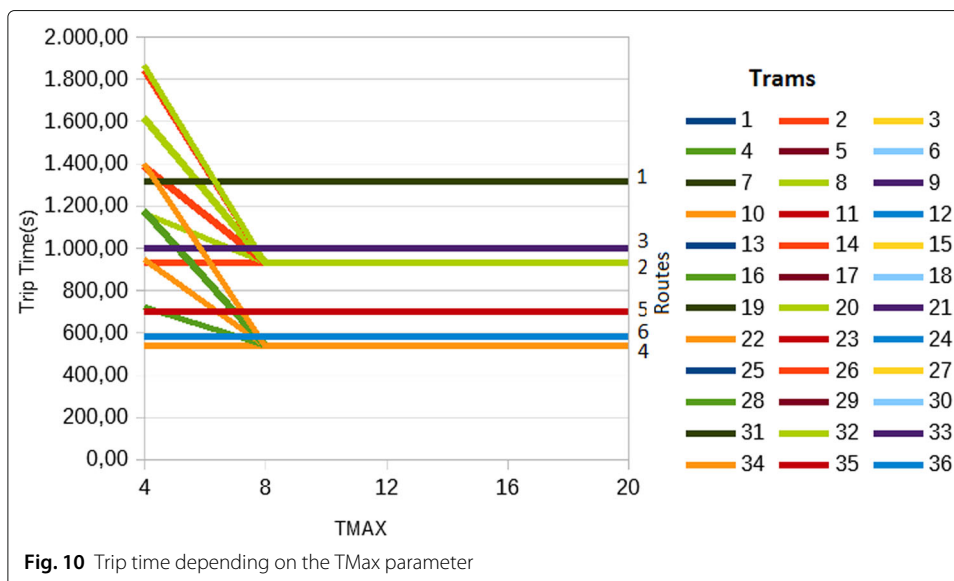


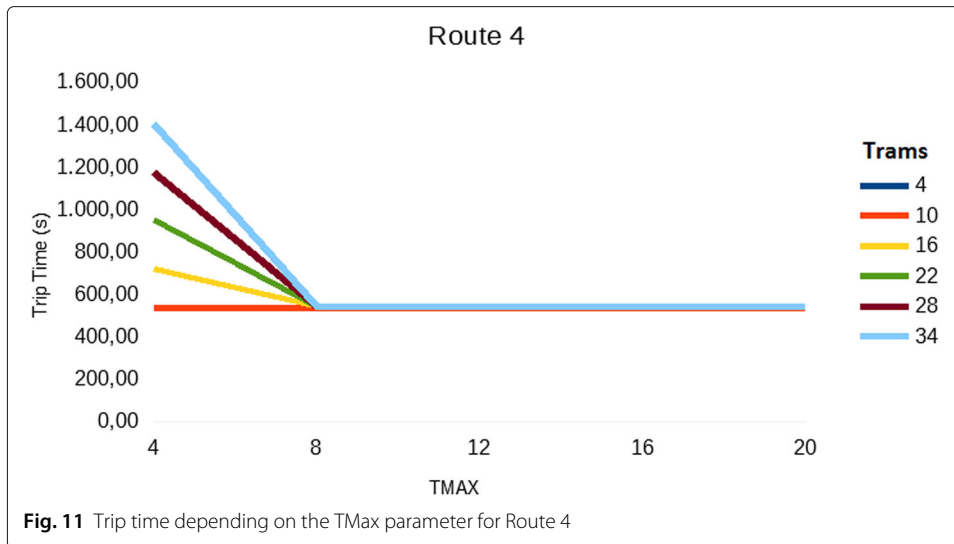


**Analysis by varying the time-out for the activation of the manual procedure**

A second series of assessments concerned the analysis of the impact on the entire traffic management of the *TMax* parameter (with  $Tmax = 4, 8, 12, 16, 20$  s), which represents the maximum time a tram can wait in front of a tram signal 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%).

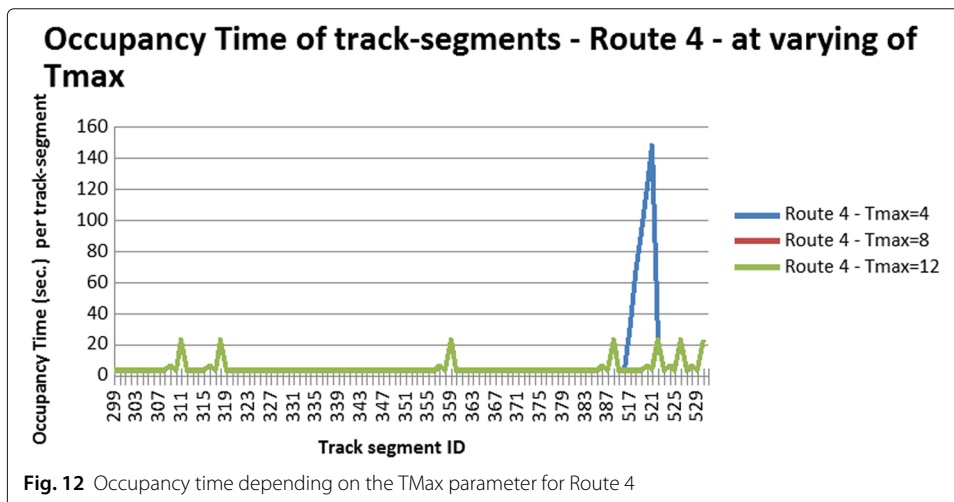
The results are presented in Fig. 10. The 6 trams on Route 4 are identified as Tram 4 (the first tram leaving from Terminal E, see Fig. 2), Tram 10, Tram 16, Tram 22, Tram 28, and Tram 34 (the last tram leaving from terminal E). Also in this case, it is possible to conclude that the *Tmax* parameter, under nominal conditions ( $tmax=8$ , or greater values), has no impact on the performance of the analyzed tramway network. This means that, in normal conditions, the activation of the manual procedure will not occur. On the other





hand, in the worst case ( $t_{max}=4$ ), we have a significant impact, mainly on routes 2 and 4. As an example, in Fig. 11, we can see the analysis results for the trams on Route 4. For  $t_{max} = 4$ , we have a total of 9 Call OCC procedure, while for the other values of  $t_{max}$  we have no call OCC procedure. A setting of  $t_{max} = 4$  actually corresponds to a quite stressful and unusual condition, since in normal situations the activation of the manual procedures should not occur.

In Fig. 12, we can see the occupancy for the segments on Route 4. The part of the track from segment 515 to 523 is very sensitive to the T<sub>max</sub> variation. Segment 515 is the point of the line where Route 4 starts sharing the track with Route 2. Results show that the first platform (track-circuit) encountered in this shared part of the track (at segment ID=522) is responsible for accumulating delays of the trams, mainly induced by the OCC calls. This means that trams arriving in the shared track from the two routes are not well distanced, they arrive at the intersection very close one each other, and while a tram occupies the track-circuit the following tram is waiting at the tram signal and has to activate the manual procedure and call OCC. This in turn may propagate the delay to the following trams, and



so on. For avoiding this problem still maintaining the  $T_{max} = 4$  value, the trams from Route 2 and 4 should be re-scheduled for reaching the intersection at different timing.

#### Analysis by varying the quality of the communication network

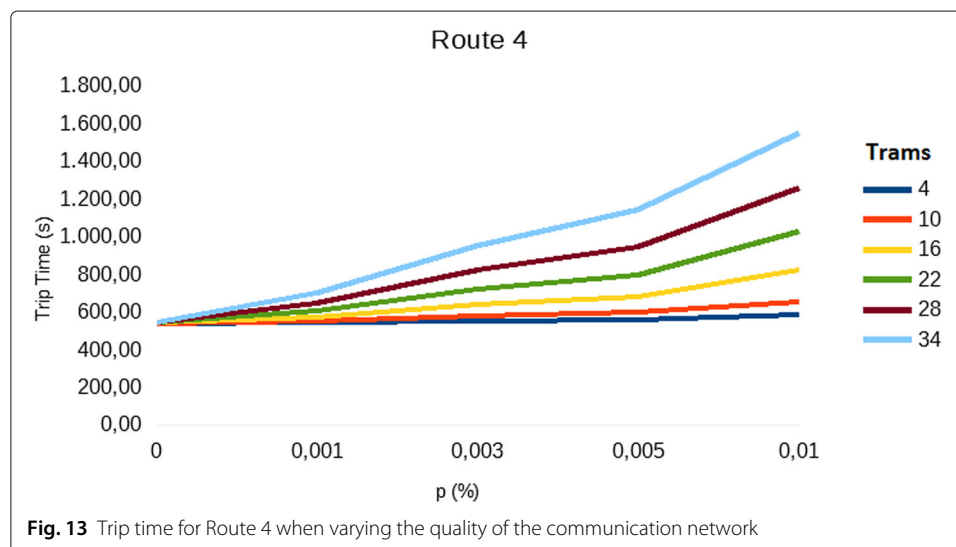
The third series of evaluations concerned the analysis of the impact of the parameter  $p$  (with  $p = 0, 0.001, 0.003, 0.005, 0.01$ ), which made it possible to understand how much the probability that a message does not arrive at its destination affects the traffic management. The lack of communication 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 tram signal set to GO. The results presented in Fig. 13, for Route 4, show that the impact of communication network quality on the performance of the analyzed tramway may be significant. The same behavior occurs for the other routes.

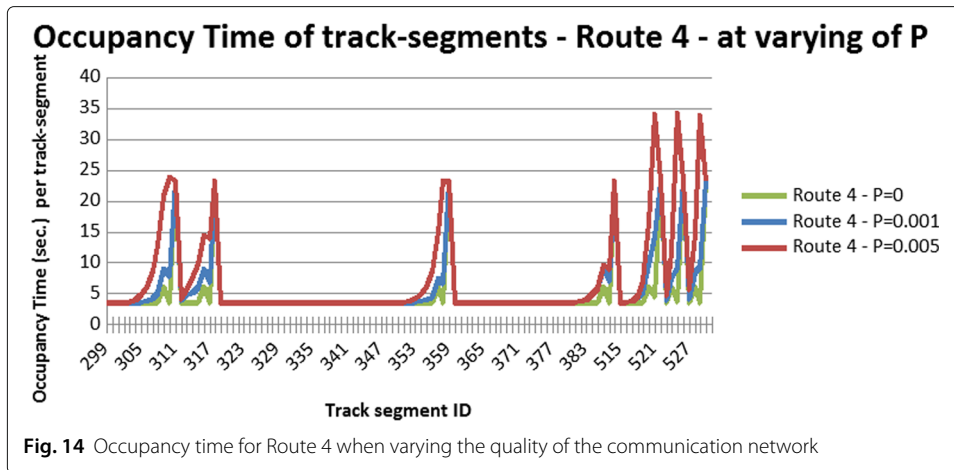
With  $p = 0.001$  (or if 1 message is lost every 1000 transmitted between OBCU and IXL), a delay is already induced on all trams, mainly to the last trams on route, with maximum peaks of 157.58 s of delay for the last tram (about 29% delay compared to the nominal travel time of 540 s). With  $p = 0.003$  (or if 3 messages are lost per 1000 transmitted), a delay is obtained on all trams, with peaks of 412.17 s of delay for the last tram (about 76% of delay compared to the nominal travel time). The results with  $p = 0.01$  are significantly worst, with an average delay of the last tram of 1010.42 s, equal to 187% of the nominal travel time. The same behavior occurs to the other routes, but Routes 2 and 4 experience greater delays than the other routes.

The impact of the network quality on the traffic, as shown in Fig. 14, is distributed throughout the whole track and the critical elements correspond to the track-circuits.

The most critical part of the route is again the one sharing the track with Route 2 (from segment ID=515).

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 s), and this delay can be propagated as a cascading effect to the following trams. Thus, this delay influences

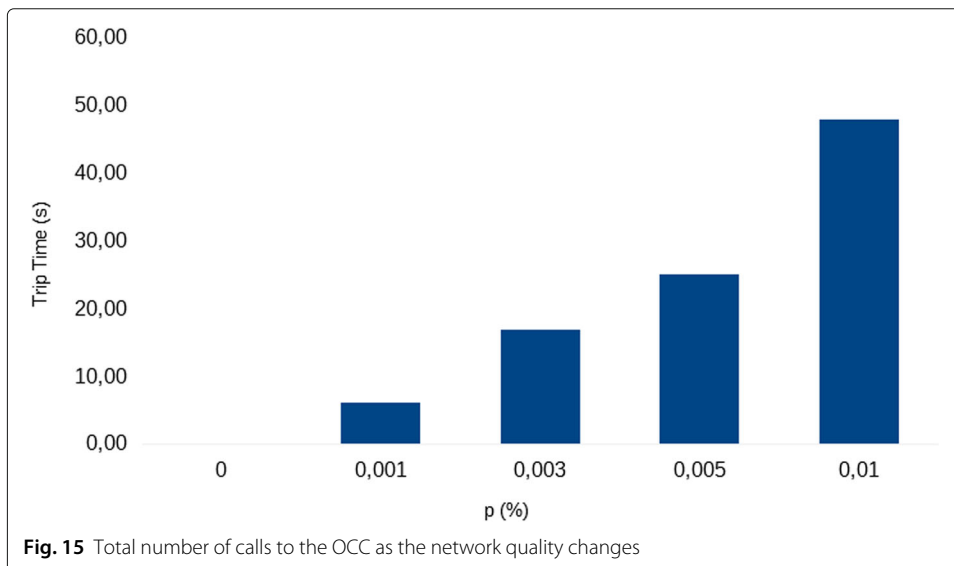


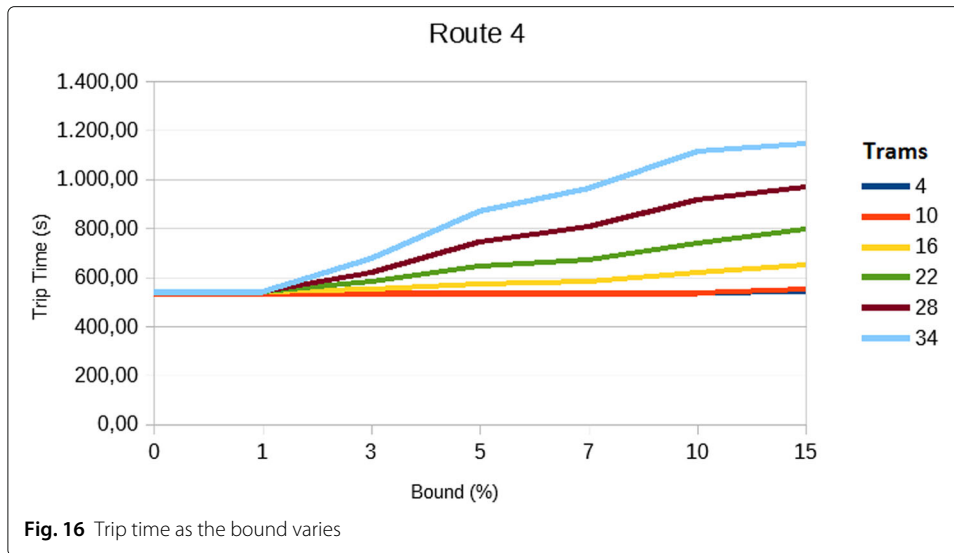


the tramway traffic of the entire area. The results are presented in Fig. 15. As expected, the accumulation of delays in the travel time shown in Fig. 13 and on the occupancy time distribution shown in Fig. 14 is proportional to the increase in the number of calls to the OCC, and therefore to the activation of the manual procedure.

**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 separated 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:  $LowerBound : (ol/AverageVelocity) * (1 - Bound)$ ;  $UpperBound : (ol/AverageVelocity) * (1 + Bound)$ .

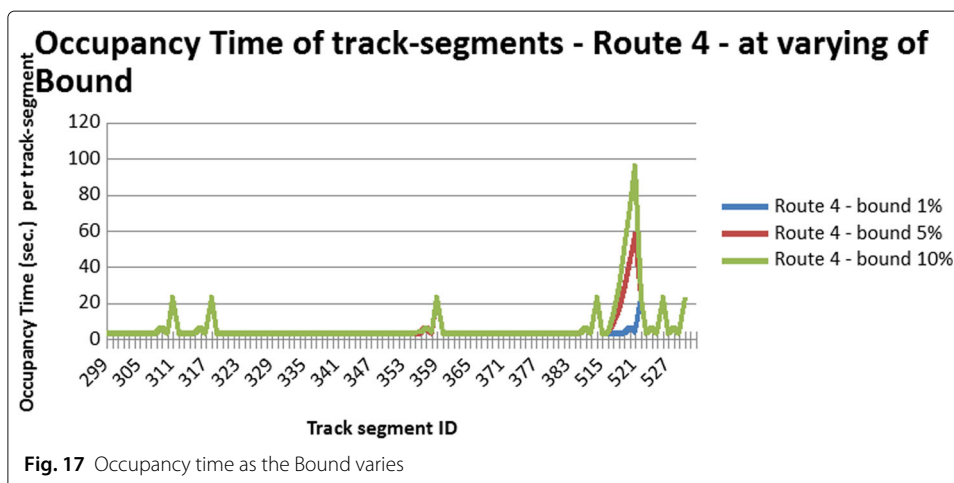


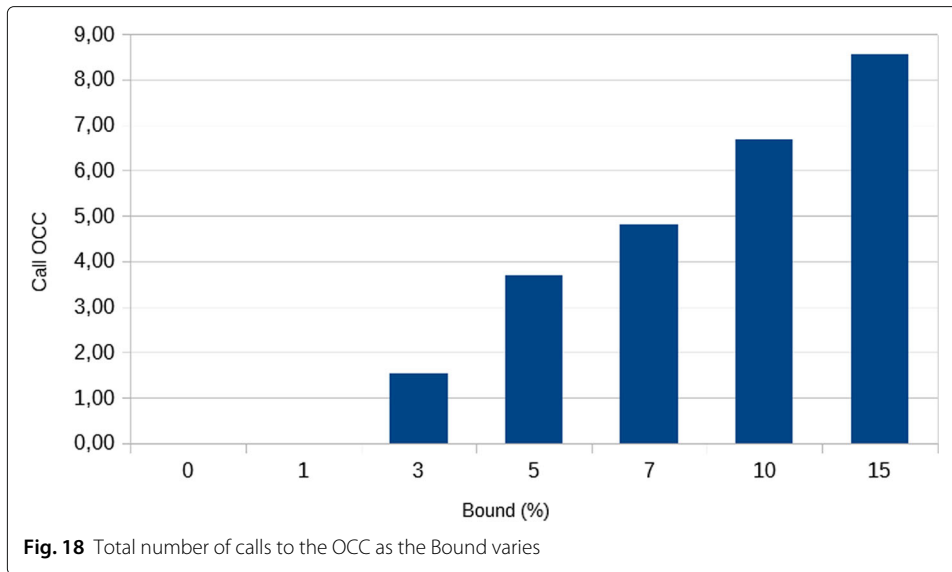


The results for Route 4 are shown in Fig. 16. As the variability of tram movement increases, trip times increase to reach a different upper limit for each tram. The worst case is that related to the last tram, whose average trip time increases to about 1143.96 s compared to the average trip time of 540 s. Therefore, we can conclude that the bound has significant cumulative impact compared to the average trip time of the trams.

As can be seen from Fig. 17, the impact of the bound does not affect the trams in the first part of the track, as the trams are well distanced by their departure time. Instead, we have the same negative impact at the intersection with Route 2 (from segment ID=515), which confirms to be a critical part of the tramway network.

Concerning the number of trams that have to activate the manual procedure with the OCC, shown in Fig. 18, we can observe that the results are consistent with that of Figs. 16 and 17, since the accumulated delays increase proportionally with the increase in the number of calls to the OCC (activations of the manual procedure).





### Combination of previous analyses

In the previous sections, we analyzed the impact of one parameter at a time concerning the metrics of interest. In this section, we consider two different parameter settings. 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 2 summarizes the setting of the parameters for the best and worst case analysis.

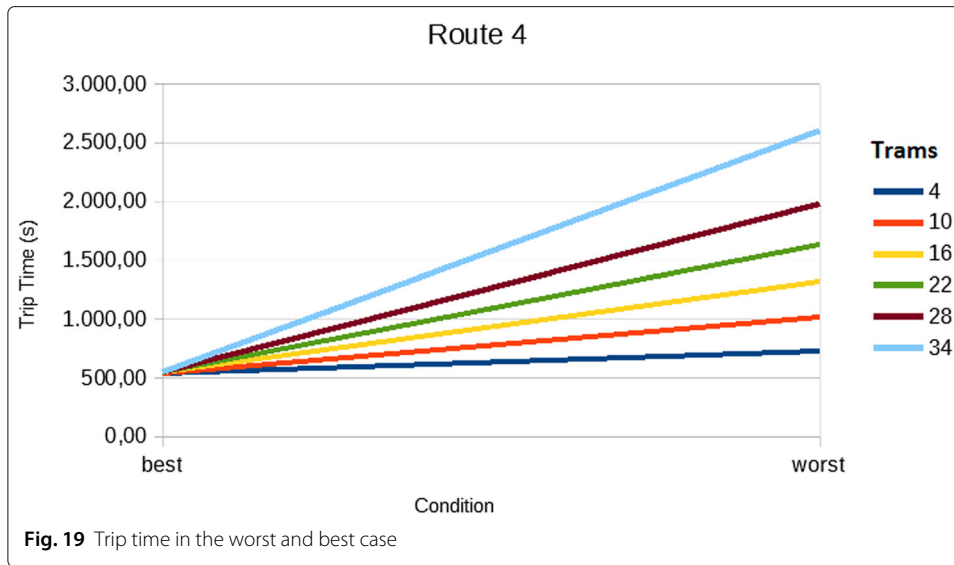
The results for Route 4 are presented in Fig. 19. Moving from the best to the worst case, we note a general increase in average trip times, which becomes more significant for the last trams. The average travel time on all trams goes from 540 s in the best case to an average of 2595 s in the worst case, which corresponds to an average delay of about 2055 s (34.25 min) per tram, or more than 380% of the total trip time for each tram.

The major negative impact is on the very first part of the track, in correspondence of the first track-circuit, but also all the following track-circuits are deeply negatively affected by the worst combination of the parameters setting (see Fig. 20). The worst conditions and more prominently the poor quality of the communication network (1% of lost messages) immediately induce a significant delay at the trams approaching the first track-circuit due to OCC calls, and then the delays are propagated to the following trams that will be queued behind the first track-circuit, and forced to call OCC as well. The same situation also happens to the following track-circuits but with less intensity, since the trams become more distanced when leaving the first track-circuit, and the negative effect of the other parameters (like the bound) are smoothed.

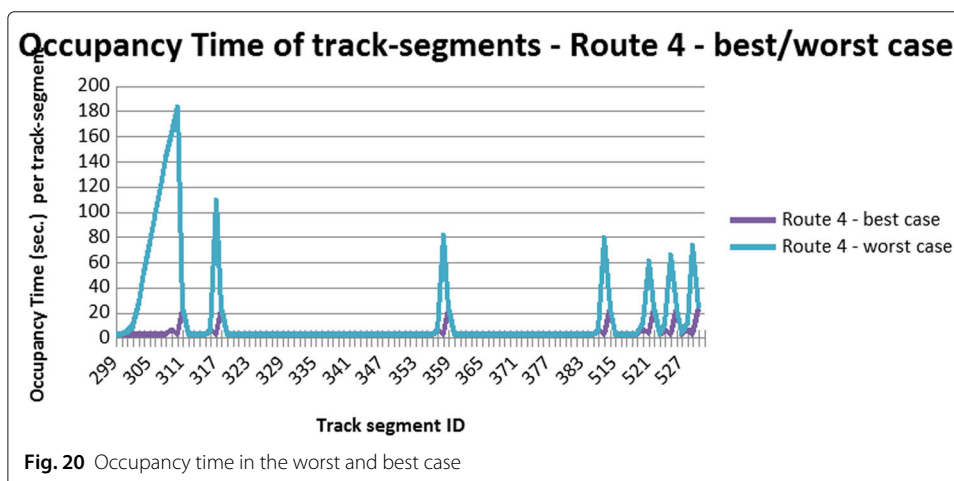
**Table 2** Parameters for the analysis in the best and worst cases

Parameter	Best value	Worst value
TMax	20 s	4 s
$p$	0	0.01
Bound	1%	15%
Epsilon	1 m	10 m





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 smaller than the sum of the individual contributions extracted respectively from Figs. 8, 10, 13, and 16. For example, summing the individual contributions of epsilon,  $TMax$ ,  $p$ , and bound for Tram 32 (last tram on Route 2) as calculated in the previous analysis, an overall delay of 3154 s is obtained (33 s of delay for  $\epsilon = 10$  m, 930 s for  $TMax = 4$  s, 1467 s for  $p = 0.01$ , 724 s for  $bound = 15\%$ ), while from Fig. 19 we can see how the overall delay in the worst case is equal to  $3529 - 933 = 2596$  s, i.e., more than 9 min less the delay calculated as a sum of individual contributions. This positive emerging behavior is due to the interdependence between the parameters and to the interdependence between the Routes 2 and 4, which share the last part of the track. The explanation is that in the worst case conditions, the negative effects induced by the worst case setting on a part of the track are actually overcome by the positive effects induced on other parts of the track, which do not occur considering the variation of one parameter in isolation. To better explain this behavior, let us see the occupancy time distribution of Route 4 in Figs. 12



and 20. The highest peak at the beginning of Route 4 in Fig. 20 (184 s), in the worst case conditions, induces at the intersection with Route 2 (from segment ID=515) a peak of about 60 s, which is significantly smaller than the peak induced in the same part of the line just considering  $T_{max}=4$  (see Fig. 12), which is about 148 s. This is a concrete example showing the usefulness of the approach for detecting complex system’s interactions.

**Analysis by varying the repair time of a tram failure**

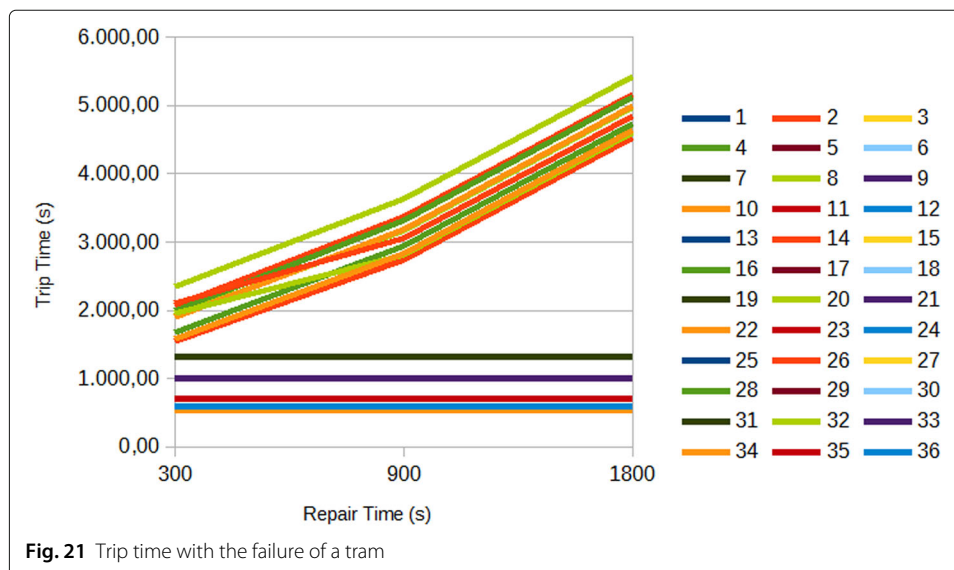
In this section, we analyze the trip time by varying the time to repair a temporary failure (outage) of a tram on the line. We simulated the presence of a failure in Tram 2 (the first tram on Route 2) starting at time 900 s (the nominal trip time for Tram 2 is about 933 s), whose effect is to stop the tram, and its repair after another 1800 s (30 min for the repair of the failure), 900 s (15 min) and 300 s (5 min). At time 900 s, Tram 2 already entered in the part of the track shared with Route 2; thus, we expect that its failure will induce some delays both in Route 4 and 2.

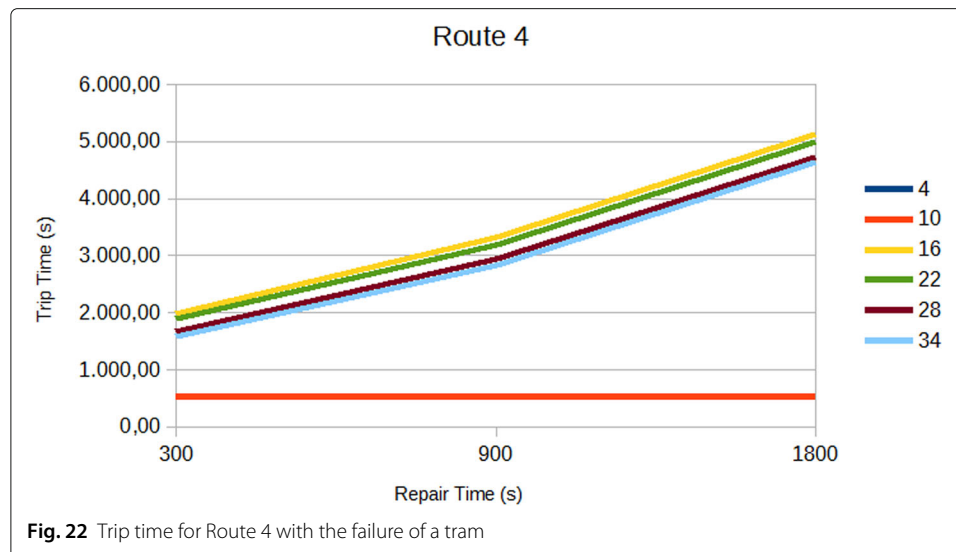
The results are presented in Fig. 21. The trams that are not directly sharing the track with Route 2 are not affected by the failure, like for all the trams on Routes 1, 3, 5, and 6. On the contrary, trams on Routes 2 and 4 are delayed. Specifically, all the trams on Route 2 are delayed due to the Tram 2 failure, since all the following trams on this route will be stopped behind him. Instead, some of the trams on Route 4 (specifically the first two trams, i.e., Tram 4 and Tram 5) are actually not affected by the failure since they already passed the segment of the track where Tram 2 is stopped, as shown in Fig. 22. The traffic congestion induced in the route increases with the outage duration.

This analysis is important as we now have multiple routes and intersections and inter-leaving of trams on different routes sharing part of the tracks, for understanding the propagation paths of the delays within the network.

**Related work**

Tramway systems, and more in general railway systems, have been studied in different contexts and with different objectives, including energy issues for reducing the





environmental impact and for minimizing the energy consumption ([5–7]), multimodal traveler information systems ([8, 9]), improving service quality ([10, 11]), reducing the impacts on users in the case of system failures ([12, 13]), rescheduling and dispatching to improve passenger comfort and convenience ([14, 15]).

On the other hand, stochastic models have been applied to several different analysis domains such as failure propagation in chemical plants [16, 17], failure in electric power systems [18, 19], performance and availability of hybrid storage systems [20], and predicting the propagation of train delays [21, 22]. Recent works on the application of the Petri Nets formalism to railway systems take into consideration specific management or decision issues such as the estimation of train delays ([23, 24]), the design of a controller for railway crossing systems ([25, 26]), train rescheduling problem ([27–30]), and performability evaluation of emergency stops for the European Rail Traffic Management System ([31]).

None of the previous works is considering the distinguishing aspects of the SISTER solutions based on the virtualization of the physical sensors. This paper builds on [4], which introduced the stochastic modeling approach for the quantitative analysis of the SISTER system, and presented the analysis of a simple tramway scenario with one line and one route only. The key extension provided in this paper is the possibility to define and analyze more complex and realistic tramways networks, with several routes and with intersections of lines. The proposed approach could also be used in synergy with other works focusing on a specific aspect of the problem whose results could be used as input in our approach. For example, the results of works dealing with tram scheduling problems could be used as input from our approach, to analyze the scenario in different tram scheduling conditions.

### 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, for capturing the key architectural components of the system and their interactions, which allows to analyze a complex multi-route network scenario. Thanks to the parametric definition of the data structures used in the model, the model can be applied to analyze different tramway topologies, with different sequences of junction areas, different numbers of routes, with possible intersections and shared tracks.

Three different metrics have been defined for analyzing the network traffic, capturing the trip time of each tram, the distribution of the traffic within each route, and the frequency of activation of the manual procedure. A sensitivity analysis has been performed by varying some key system parameters, including the accuracy of the on-board positioning system, the time-out for the activation of manual procedures, and the quality of the OBCU-IXL communication network. Furthermore, we analyzed the impact of the level of interference between the mobility of the trams and the surrounding environment, as well as the effect of possible failures that can temporarily block the travel of a tram and congest the traffic of the tramway network.

The modeling approach has been applied for analyzing an existing Chinese tramway network, and the results allowed to identify the parts of the network that were more sensible and then more critical to the variations of the parameters. It was possible to conclude that in nominal traffic conditions (no problem of loss of messages, no tram affected by failure) the accuracy has no impact on the performance, while the other parameters affect the journey of the trams in different ways. The delays induced by decreasing the quality of the communication network are almost equally distributed towards the route in correspondence of the track-circuits. On the contrary, shorting the time-out parameter or increasing the variability of the tram movement induces the major delays in the parts of the tracks shared between different routes.

#### **Acknowledgements**

This work has been developed in the context of the REGIONE TOSCANA POR FESR 2014-2020 SISTER "Signaling & Sensing Technologies in Railway application", and by the European Union's Horizon 2020 research and innovation program.

#### **Authors' contributions**

All the author contributed equally to this work.

#### **Funding**

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 Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 823788.

#### **Availability of data and materials**

Not applicable.

#### **Competing interests**

The authors declare that they have no competing interests.

#### **Author details**

<sup>1</sup>Computing Institute, Federal University of Alagoas, Maceió, Brazil. <sup>2</sup>Resiltech S.R.L., Pontedera, Italy. <sup>3</sup>Dept. of Math. and Computer Science, University of Florence, Firenze, Italy. <sup>4</sup>Consorzio Interuniversitario Nazionale per l' Informatica (CINI), University of Florence, Firenze, Italy. <sup>5</sup>Innovation & Technology, Thales Italia SpA, Pontedera, Italy.

Received: 11 May 2020 Accepted: 24 January 2021

Published online: 12 February 2021

## References

1. Sanders WH, Meyer JF (2001) Stochastic activity networks: formal definitions and concepts (Brinksma E, Hermanns H, Katoen J-P, eds.), Vol. 2090. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/3-540-44667-2\\_9](https://doi.org/10.1007/3-540-44667-2_9)
2. Sanders WH (1999) Integrated frameworks for multi-level and multi-formalism modeling. In: Proceedings 8th International Workshop on Petri Nets and Performance Models (Cat. No.PR00331). pp 2–9. <https://doi.org/10.1109/PNPM.1999.796527>
3. Sanders WH, Courtney T, Deavours D, Daly D, Derisavi S, Lam V (2003) Multi-formalism and multi-solution-method modeling frameworks: The Möbius approach. In: Proceedings of the Symposium on Performance Evaluation - Stories and Perspectives. pp 241–256
4. da Silva LD, Mongelli D, Lollini P, Bondavalli A, Mandò G (2019) Performability analysis of a tramway system with virtual tags and local positioning. In: 2019 9th Latin-American Symposium on Dependable Computing (LADC). pp 1–10. <https://doi.org/10.1109/LADC48089.2019.8995712>
5. Pugi L, Grasso F, Rossi G (2018) 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). pp 1–6. <https://doi.org/10.1109/EEEIC.2018.8494431>
6. Yan Z, Min Y (2018) Energy consumption prediction of trams based on grey relational analysis and regression model. In: 2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE). pp 111–117. <https://doi.org/10.1109/IRCE.2018.8492926>
7. Kubín J, Ferková Z (2015) Influence of driving style of a tram driver on the tram's energy consumption. In: 2015 International Conference on Electrical Drives and Power Electronics (EDPE). pp 417–421. <https://doi.org/10.1109/EDPE.2015.7325331>
8. Dib O, Manier M-A, Moalic L, Caminada A (2017) A multimodal transport network model and efficient algorithms for building advanced traveler information systems. *Transp Res Procedia* 22:134–143. <https://doi.org/10.1016/j.trpro.2017.03.020>. 19th EURO Working Group on Transportation Meeting, EWGT2016, 5-7 September 2016, Istanbul, Turkey
9. Dib O, Manier M, Moalic L (2016) Advanced modeling approach for computing multicriteria shortest paths in multimodal transportation networks. In: 2016 IEEE International Conference on Intelligent Transportation Engineering (ICITE). pp 40–44. <https://doi.org/10.1109/ICITE.2016.7581304>
10. dell'Olio L, Ibeas A, Cecin P (2011) The quality of service desired by public transport users. *Transp Policy* 18(1):217–227. <https://doi.org/10.1016/j.tranpol.2010.08.005>
11. Pagliara F, Biggiero L (2017) Some evidence on the relationship between social exclusion and high speed rail systems. *HKIE Trans* 24(1):17–23. <https://doi.org/10.1080/1023697X.2016.1263162>
12. Consilvio A, Di Febbraro A, Sacco N (2016) Stochastic scheduling approach for predictive risk-based railway maintenance. In: 2016 IEEE International Conference on Intelligent Rail Transportation (ICIRT). pp 197–203. <https://doi.org/10.1109/ICIRT.2016.7588732>
13. D'Acerno L, Placido A, Botte M, Montella B (2016) A methodological approach for managing rail disruptions with different perspectives. *Int J Math Models Methods Appl Sci* 10:80–86
14. Gao Y, Kroon L, Schmidt M, Yang L (2016) Rescheduling a metro line in an over-crowded situation after disruptions. *Transp Res B Methodol* 93:425–449. <https://doi.org/10.1016/j.trb.2016.08.011>
15. Zhan S, Zhao J, Peng Q (2016) Real-time train rescheduling on highspeed railway under partial segment blockages. *J China Railw Soc* 38:1–13
16. Sierra D, Montecchi L, Mura I (2019) Stochastic modeling and analysis of vapor cloud explosions domino effects in chemical plants. *J Braz Comput Soc* 25(1):1–19. <https://doi.org/10.1186/s13173-019-0092-8>
17. Sierras D, Briceño J, Buitrago H, Rozo B, Montecchi L, Mura I (2018) Probabilistic modeling of failure domino effects in chemical plants. In: 2018 Eighth Latin-American Symposium on Dependable Computing (LADC). pp 57–66. <https://doi.org/10.1109/LADC.2018.00016>
18. Chiaradonna S, Di Giandomenico F, Masetti G (2016) Analyzing the impact of failures in the electric power distribution grid. In: 2016 Seventh Latin-American Symposium on Dependable Computing (LADC). pp 99–108. <https://doi.org/10.1109/LADC.2016.23>
19. Chiaradonna S, Di Giandomenico F, Lollini P (2011) Definition, implementation and application of a model-based framework for analyzing interdependencies in electric power systems. *Int J Crit Infrastruct Prot* 4(1):24–40. <https://doi.org/10.1016/j.jicp.2011.03.001>
20. Borba E, Tavares E (2017) Stochastic modeling for performance and availability evaluation of hybrid storage systems. *J Syst Softw*. 134(C):1–11. <https://doi.org/10.1016/j.jss.2017.08.043>
21. Corman F, Kecman P (2018) Stochastic prediction of train delays in real-time using Bayesian networks. *Transp Res C Emerg Technol* 95:599–615. <https://doi.org/10.1016/j.trc.2018.08.003>
22. Bavafa-Toosi Y, Blendinger C, Mehrmann V, Steinbrecher A, Unger R (2008) A new methodology for modeling, analysis, synthesis, and simulation of time-optimal train traffic in large networks. *IEEE Trans Autom Sci Eng* 5(1):43–52. <https://doi.org/10.1109/TASE.2007.897613>
23. Ricci S (2009) The use of petri nets models in railway traffic applications. *IFAC Proc Vol* 42(5):151–156. <https://doi.org/10.3182/20090610-3-IT-4004.00031>. 2nd IFAC Workshop on Dependable Control of Discrete Systems
24. Milinković S, Marković M, Vesković S, Ivić M, Pavlović N (2013) A fuzzy petri net model to estimate train delays. *Simul Model Pract Theory* 33:144–157. <https://doi.org/10.1016/j.simpat.2012.12.005>. EUROSIM 2010
25. Ahmad F, Khan SA (2013) Specification and verification of safety properties along a crossing region in a railway network control. *Appl Math Model* 37(7):5162–5170. <https://doi.org/10.1016/j.apm.2012.10.047>
26. Liu B, Ghazel M, Toguyéni A (2016) Model-based diagnosis of multi-track level crossing plants. *IEEE Trans Intell Transp Syst* 17(2):546–556. <https://doi.org/10.1109/TITS.2015.2478910>
27. Giglio D, Sacco N (2016) A petri net model for analysis, optimisation, and control of railway networks and train schedules. In: 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). pp 2442–2449. <https://doi.org/10.1109/ITSC.2016.7795949>
28. HIRAI C, KUNIMATSU T, TOMII N, KONDOU S, TAKABA M (2009) A train stop deployment planning algorithm using a petri-net-based modelling approach. *Q Rep RTRI* 50(1):8–13. <https://doi.org/10.2219/rtrqr.50.8>

29. Wang P, Ma L, Goverde RMP, Wang Q (2016) Rescheduling trains using petri nets and heuristic search. *IEEE Trans Intell Transp Syst* 17(3):726–735. <https://doi.org/10.1109/TITS.2015.2481091>
30. Enache MF, Letia TS (2019) Approaching the railway traffic resilience with object enhanced time petri nets. In: 2019 23rd International Conference on System Theory, Control and Computing (ICSTCC). pp 338–343. <https://doi.org/10.1109/ICSTCC.2019.8885878>
31. Biagi M, Carnevali L, Paolieri M, Vicario E (2017) Performability evaluation of the ERTMS/ETCS – Level 3. *Transp Res C Emerg Technol* 82:314–336. <https://doi.org/10.1016/j.trc.2017.07.002>

### **Publisher's Note**

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

---

Submit your next manuscript at ▶ [springeropen.com](https://www.springeropen.com)

---