Dipartimento di Informatica
Università del Piemonte Orientale "A. Avogadro"
Viale Teresa Michel 11, 15121 Alessandria
http://www.di.unipmn.it


# SAN models of communication scenarios inside the Electrical Power System

*Authors: Daniele Codetta-Raiteri, Roberto Nai*
*(daniele.codetta_raiteri@mfn.unipmn.it, robertonai@libero.it)*

TECHNICAL REPORT TR-INF-2009-07-05-UNIPMN
*(July 2009)*

The University of Piemonte Orientale Department of Computer Science Research Technical Reports are available via WWW at URL `http://www.di.unipmn.it/`.

Plain-text abstracts organized by year are available in the directory

## Recent Titles from the TR-INF-UNIPMN Technical Report Series

2009-04 *On-line Product Conguration using Fuzzy Retrieval and J2EE Technology*, Portinale, L., Galandrino, M., May 2009.

2009-03 *GSPN Semantics for Continuous Time Bayesian Networks with Immediate Nodes*, Portinale, L., Codetta-Raiteri, D., March 2009.

2009-02 *The TAAROA Project Specification*, Anglano, C., Canonico, M., Guazzone, M., Zola, M., February 2009.

2009-01 *Knowledge-Free Scheduling Algorithms for Multiple Bag-of-Task Applications on Desktop Grids*, Anglano, C., Canonico, M., February 2009.

2008-09 *Case-based management of exceptions to business processes: an approach exploiting prototypes*, Montani, S., December 2008.

2008-08 *The ShareGrid Portal: an easy way to submit jobs on computational Grids*, Anglano, C., Canonico, M., Guazzone, M., October 2008.

2008-07 *BuzzChecker: Exploiting the Web to Better Understand Society*, Furini, M., Montangero, S., July 2008.

2008-06 *Low-Memory Adaptive Prefix Coding*, Gagie, T., Nekrich, Y., July 2008.

2008-05 *Non deterministic Repairable Fault Trees for computing optimal repair strategy*, Beccuti, M., Codetta-Raiteri, D., Franceschinis, G., July 2008.

2008-04 *Reliability and QoS Analysis of the Italian GARR network*, Bobbio, A., Terruggia, R., June 2008.

2008-03 *Mean Field Methods in performance analysis*, Gribaudo, M., Telek, M., Bobbio, A., March 2008.

2008-02 *Move-to-Front, Distance Coding, and Inversion Frequencies Revisited*, Gagie, T., Manzini, G., March 2008.

2008-01 *Space-Conscious Data Indexing and Compression in a Streaming Model*, Ferragina, P., Gagie, T., Manzini, G., February 2008.

2007-05 *Scheduling Algorithms for Multiple Bag-of-Task Applications on Desktop Grids: a Knowledge-Free Approach*, Canonico, M., Anglano, C., December 2007.

2007-04 *Verifying the Conformance of Agents with Multiparty Protocols*, Giordano, L., Martelli, A., November 2007.

2007-03 *A fuzzy approach to similarity in Case-Based Reasoning suitable to SQL implementation*, Portinale, L., Montani, S., October 2007.

2007-02 *Space-conscious compression*, Gagie, T., Manzini, G., June 2007.

2007-01 *Markov Decision Petri Net and Markov Decision Well-formed Net Formalisms*, Beccuti, M., Franceschinis, G., Haddad, S., February 2007.

2006-03 *New challenges in network reliability analysis*, Bobbio, A., Ferraris, C., Terruggia, R., November 2006.

# Contents

# SAN models of communication scenarios inside the Electrical Power System

Daniele Codetta-Raiteri, Roberto Nai
Dipartimento di Informatica, Università del Piemonte Orientale
Viale T. Michel 11, 15121 Alessandria, Italy
*e-mail*: raiteri@mfn.unipmn.it, robertonai@libero.it

**Abstract**

This report provides all the details about the models and the quantitative results presented in [1], about the simulation of communication scenarios inside the Electrical Power System. In particular, the scenarios deal with the communication between one area control centre and a set of substations in a distribution grid, exchanging commands and signals by means of a redundant communication network. The communication may be affected by threats such as the communication network failure, or intrusions into the communication, causing the loss of commands or signals. The scenarios have been modeled and simulated in form of Stochastic Activity Networks, with the purpose of evaluating the effects of such threats on the communication reliability.

*Acronym list:*

| | |
|---|---|
| EPS | Electrical Power System |
| kbps | kilobit per second |
| SAN | Stochastic Activity Network |
| SPN | Stochastic Petri Net |

## 1 Introduction

This work was developed inside the EU funded project named CRUTIAL (*CRitical UTility InfrastructurAL resilience*) [2] and addressing the resilience of the Electrical Power System (EPS). This system can be structured in three subsystems (Fig. 1):

- the *Power generation* consists of the set of plants generating the electric power;
- the *Transmission grid* is the set of high voltage (HV) electric lines, substations and control centres necessary to transport the electric power from the power plants to the distribution grid of each region of the territory;
- the *Distribution grid* is the set of medium (MV) or low voltage (LV) electric lines, substations and control centres in charge of transporting the electric power from the transmission grid to the consumers located in a certain region.

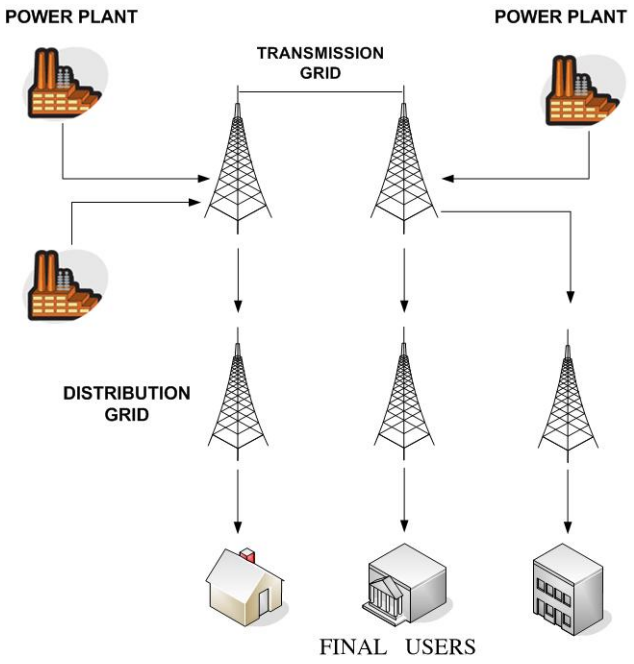


Figure 1: The general EPS architecture.

In this report, we take into account some of the critical scenarios defined in [3] and dealing with the communication between a control centre and a set of substations inside a distribution grid. This communication consists of the delivery of commands from the control centre to the substations, and the delivery of signals from the substations to the control centre. This communication is performed through the Internet and the aim of the scenarios is investigating the effects of several threats to the delivery of commands or signals. For instance, an attacker may perform an intrusion in the communication with the possibility of sending fake commands to the substations with the aim of causing their malfunctioning. In another scenario, the failure of the communication network may compromise the exchange of data between the control centre and the substations. In this report, we model and simulate the scenarios under exam in form of *Stochastic Activity Networks* (SAN) [4], a particular form of Stochastic Petri Net (SPN) [5]; the purpose is evaluating the effect of failures and attacks to the communication between the control centre and the set of substations, in terms of probability and quantity of lost data. The design, the composition and the simulation of the SAN models have been performed by means of the *Möbius* tool [6, 7].

The report is structured as follows: Sec. 2 describes the case study under exam specifying the communication protocol between the involved sites in case of transmission of commands and signals, together with the possible threats to the communication in each scenario. Sec. 3 provides some general notions about the SAN formalism, while in Sec. 4, the SAN models of the several aspect of the case study are described in detail;

then, such models are composed in order to obtain the models of the scenarios. The results of the model simulation are reported in Sec. 5.

# 2 The case study

The scenarios under exam [3] deal with the communication between a control centre and a set of substations inside a distribution grid, with the aim of sending commands from the control centre to the substations, and signals from the substations to the control centre (Fig. 2). Such transmissions are performed by means of a redundant communication network.

Typically a substation is connected to several electrical lines for the electrical power transportation, and executes the commands coming from the control centre. Such commands usually concern some operations to be performed to the electrical lines. In the case of the distribution grid, the same command may be sent to all the substations, for instance, an arming or disarming command [3]. The generation of a command by the control centre is a stochastic event occurring as a consequence of a command coming from the transmission grid, or as a consequence of the state of the distribution grid described by the signals coming from the substations.

The signals sent from the substations to the control centre may describe the state of the substation or the state of the electrical lines connected to the substation. Such information are useful to the control centre in order to monitor the state of the portion of the distribution grid under its control. Signals are periodically transmitted from the substations to the control centre.

So, the communication of commands and signals is fundamental for the correct functioning of the EPS. Anyway, such communication may be affected by threats such as attacks and failures. The aim of the scenarios under study, is evaluating their impact on the communication reliability.

## 2.1 Command and signal sessions

In our case study, we suppose that each command generated by the control centre has to be executed by all the substations; therefore, a copy of the command is sent to each substation. Moreover, we assume that the execution of command by a substation is notified to the control centre by the transmission of an acknowledgment coming from the substation. So, the generation, the transmission and the execution of a command are performed according to the following sequence of operations that we call "command session":

1. the control centre opens the command session: it generates the command and starts collecting the acknowledgments coming from the substations and concerning the command execution, until a certain time out expires;

2. a copy of the command is transmitted on the available communication network to each substation;

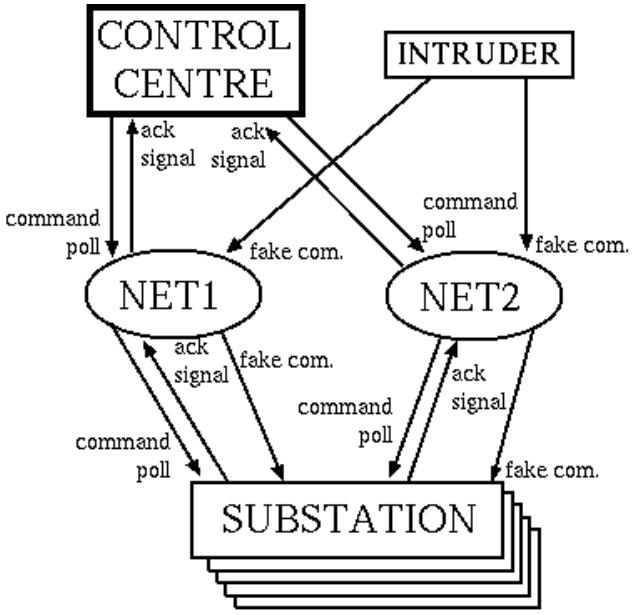


Figure 2: The scheme of the case study.

3. each substation executes the command and generates an acknowledgment proving the execution of the command;

4. each acknowledgment is transmitted on the available communication network to the control centre;

5. the time out for the acknowledgments collection expires and the command session is closed.

If the number of substations is $N$, a command session is successful if at least $N - 1$ acknowledgments are received by the control centre before that the time out expires. If instead, more than one acknowledgment is missing when the time out expires, then the command session is considered to be failed. In the case study investigated in this report, we suppose that $N$ is equal to 10 (10 substations are present in the case study). We assume that at most one command session is running at a certain time, so parallel command sessions are not possible.

In our case study, the time for an event to occur can be a deterministic time or a random time; in the first case, the time to occur is a constant, while in the second case, such time is a random variable ruled by a *negative exponential distribution* whose *rate* is equal to the inverse value of the *mean time* for the event to occur. Let us consider the command session: the time for the control centre to generate a command is random and is equal to 6 $h$ on average (the mean time between two command sessions is 6 $h$ on average). The transmission of a command (or any other kind of packet) by the communication network takes 1 *sec.* on average. The time to execute a command and generate the corresponding acknowledgment by the substation,

takes 1 *sec.* on average. The time out for the acknowledgments collection by the control centre, is exactly 20 *sec.*

In the case study investigated in this report, we suppose that signals are not sent by a substation in an autonomous way, but we assume that they are generated as a reply to a poll request: periodically the control centre polls all the substations by sending a poll request to each of them, and they reply by sending a signal to the control centre.

The protocol for the communication of signals is similar to the case of the communication of commands: we call "signal session" the following sequence of operations:

1. the control centre opens the signals sessions: it generates a poll and starts collecting signals coming from the substations, until a certain time out expires;
2. a poll request is transmitted on the available communication network to each substation;
3. each substation generates the signal;
4. each signal is transmitted on the available communication network to the control centre;
5. the time out for the signals collection expires and the signal session is closed.

A signal session is successful if at least $N-1$ signals are collected by the control centre before the time out expiration, where $N$ is the number of substations ($N = 10$). So, if at least two signals are missing when the time out expires, then the signal session is considered to be failed. We assume that at most one signal session is running at a certain time, so parallel command sessions are not possible. A signal session may be running in parallel with a command session.

The time between the closure of a signal session and the opening of the next one, is exactly 5 *min.*, while the time for the substation to generate the signal is 1 *sec.* on average. The time out for the signals arrival at the control centre is exactly 20 *sec.* The occurrence (mean) times for the events in a command or signal session, are summarized in Table 1.

### 2.1.1 Packets transmission

In our case study, the transmission of the command copies, the acknowledgments, the poll requests and the signals, is performed according to the Internet protocols assuming that two communication networks ($NET1$ and $NET2$) can be exploited to transmit.

$NET1$ is usually used for the communication between the control centre and the substations. We suppose that the bandwidth of each communication network is equal to 16 *kbps* and that the transmission of each packet consumes 1 *kbps* of the bandwidth. This means that no more than 16 packets can be transmitted on the same communication network at the same time. For example, if 10 signals are generated at the same time,

| Event | Type of event | (mean) time to occur | occurring rate |
|---|---|---|---|
| command generation | stochastic | 6.000E+00 $h$ | 1.667E-1 $h^{-1}$ |
| command execution | stochastic | 2.778E-04 $h$ | 3.600E+3 $h^{-1}$ |
| time out for ack. | deterministic | 5.556E-03 $h$ | - |
| poll generation | deterministic | 8.333E-02 $h$ | - |
| signal generation | stochastic | 2.778E-04 $h$ | 3.600E+3 $h^{-1}$ |
| time out for signals | deterministic | 5.556E-03 $h$ | - |
| packet transmission | stochastic | 2.778E-04 $h$ | 3.600E+3 $h^{-1}$ |

Table 1: The (mean) occurrence time (and the corresponding rates) for the events in a command or signal session.

they will consume 10 *kbps* of the bandwidth of $NET1$: during their transmission, the bandwidth of $NET1$ available for the eventual transmission of other packets is 6 *kbps*. When $NET1$ completes the transmission of the signals, the available bandwidth of $NET1$ is 16 *kbps* again. But, it may happen that the current available bandwidth of $NET1$ is not enough to transmit all the packets. In this case, $NET2$ is used to transmit the packets that $NET1$ can not transmit. This may happen if a command session and a signal session are running in parallel way. For instance, let us assume that the bandwidth of $NET1$ is currently consumed to transmit 10 signals, and before their transmission is complete, 10 acknowledgments are generated. In this case, 6 acknowledgments will be transmitted by $NET1$ with the consequent complete consume of its bandwidth, while the remaining 4 acknowledgments will be directed to $NET2$ for the transmission. Another situation where $NET2$ can be exploited for transmission is the case of the failure of $NET1$, as described in Sec. 2.2.2.

Actually, we could have specified that the transmission of a packet requires less than 1 *kbps* of the bandwidth, or that a communication network has a bandwidth higher than 16 *kbps*; in this way, the communication network would be able to transmit more than 16 packets at the same time. Our choice depends on the fact that one of the goals of the scenarios is evaluating the effect of the bandwidth consumption to the communication reliability. To this aim, if the communication networks had an higher transmission capacity, then we would need to consider more than 10 substations in the scenarios, eventually making the simulation computational costs worse.

## 2.2 Threats to the communication

The communication between the control centre and the substations may be affected by several threats causing the loss of commands, acknowledgments, poll or signals, and therefore determining the failure of command or signal sessions. In this report, we focus on the possibility of intrusions into the communication between the sites, where the intruder can send fake commands to the substations causing their temporary malfunctioning. Moreover, we focus on the possibility of temporary failure of the communication network $NET1$ or $NET2$, with the consequent temporary impossibility to transmit on the failed network.

| Event | mean time to occur | occurring rate |
|---|---|---|
| intrusion occurrence | 720 $h$ | 0.00139 $h^{-1}$ |
| intrusion duration | 5 $h$ | 0.2 $h^{-1}$ |
| fake command generation | 1 $h$ | 1.0 $h^{-1}$ |
| substation recovery | 12 $h$ | 0.08333 $h^{-1}$ |

Table 2: The mean occurrence time and the corresponding rates about the events in the intrusion. The probability of execution of a fake command by the substation is 0.01.

#### 2.2.1 Intrusion and transmission of fake commands

We suppose that attackers may succeed in performing intrusions into the communication between the control centre and the substations: during an intrusion and by exploiting $NET1$ or $NET2$, the attacker may send several fake commands to the substations pretending to be the control centre. We assume that each fake command is sent to all the substations, and each substation is able to distinguish the fake commands from the original ones: if such protection is successful the fake command is discarded, but it may happen that the protection fails and as a consequence, the fake command is executed by the substation. This determines the temporary failure of the substation functions, so during the period of unavailability, the substation does not react to the command copies or poll requests received from the control centre (or to eventual other fake commands). This may lead to the failure of command or signal sessions because the control centre will not receive any acknowledgment or signal by the unavailable substation. The substation unavailability is temporary: the failure of the substation due to the execution of a fake command, can be recovered, so the substation can turn available again replying to commands and polls. A fake command may be generated while a command session, a signal session, or both are running.

We suppose that an intrusion occurs every month on average. During an intrusion, a fake command is sent by the attacker every hour on average. The duration of an intrusion is 5 $h$ on average. The mean time to recover the substation from the execution of a fake command, is 12 $h$. Such times and the corresponding rates are summarized in Table 2. The probability that the substation protection discards a fake command is 0.99, so the probability that a substation executes a fake command is 0.01. We assume that no more than one intrusion can be running at the same time.

#### 2.2.2 Communication Network failure

The communication between the control centre and the substations is allowed by the communication networks (*NET1* and *NET2*). Each of them may become unavailable due to its own failure; we assume that such unavailability is temporary because the failure can be recovered. Since $NET1$ and $NET2$ are redundant, if $NET1$ network is not available, then the $NET2$ can be used for the transmission among the sites. Since a communication network can not transmit more than 16 packets at the same time (Sec. 2.1.1), if only one

| Component | MTTF | Failure Rate | MTTR | Repair Rate |
|---|---|---|---|---|
| $NET1$ | 720 $h$ | 0.00139E-3 $h^{-1}$ | 12 $h$ | 0.08333 $h^{-1}$ |
| $NET2$ | 720 $h$ | 0.00139E-3 $h^{-1}$ | 12 $h$ | 0.08833 $h^{-1}$ |

Table 3: The mean time to fail (MTTF), the failure rate, the mean time to repair (MTTR) and the repair rate of the communication networks.

network is available at a certain time, its bandwidth may not be enough to transmit all the packets; this may happen if a command session is running in parallel with a signal session, or if a fake command has been generated during a command or signal session. In this case, some packets may be lost. For instance, if $NET1$ is failed, $NET2$ is working, its bandwidth is currently used to transmit 10 signals, and 10 acknowledgments are generated in the meanwhile, then 6 acknowledgments will consume the available bandwidth of $NET2$ and the remaining 4 acknowledgments can not be transmitted and they will be lost.

Besides the command and signal sessions, the failure of the communication network $NET1$ or $NET2$ may compromise the transmission of fake commands as well. The communication between the control centre and the substation becomes impossible if both $NET1$ and $NET2$ are unavailable at the same time. The mean time to failure and the mean time to repair of $NET1$ and $NET2$ are reported in Table 3.

### 2.3 Scenarios definition

In this report, we are interested to evaluate the case study in three scenarios:

- Scenario 1: the communication network failures may not occur, but the intrusions may occur;
- Scenario 2: the communication network failures may occur, but the intrusions may not occur;
- Scenario 3: both the communication network failures and the intrusions may occur.

## 3 Basic notions about the SAN formalism

SANs can be considered as a particular form of Stochastic Petri Nets (SPN) [5]. So, a SAN model is characterized by *places*, each containing a certain number of *tokens* (*marking*). A place graphically appears as a circle. In the SAN formalism, there are no coloured tokens [8, 9]. The marking of a certain set of places enables the completion (firing) of *activities* (transitions) whose effect is modifying in some way the marking of the places. Activities graphically appear as vertical bars.

In the SAN formalism, the completion of an activity can be instantaneous (immediate) or timed. In the second case, the completion time can be a constant value or a random value. If the completion time is random, its value has to be ruled by a probability distribution; in this report, we always resort to the negative exponential one, but several other distributions are available in the SAN formalism. Moreover, in

this report, we call "immediate activity" an activity completing as soon as it is enabled; we call "deterministic activity" an activity whose time to complete is deterministic and not immediate; we call "stochastic activity" an activity whose time to complete is a random variable ruled by the negative exponential distribution.

The completion of an activity of any kind is enabled by a particular marking of a set of places. Such marking can be expressed by connecting the activity to the places by means of oriented arcs, as it is possible in SPNs. The effect of the activity completion can be specified in the same way. Another way to express the marking enabling a certain activity consists of using *input gates*. An input gate is connected to an activity and to a set of places; the input gate is characterized by two expressions:

- a *predicate* consists of a Boolean condition expressed in terms of the marking of the places connected to the gate; if such condition holds, then the activity connected to the gate is enabled to complete.
- a *function* expresses the effect of the activity completion on the marking of the places connected to the gate.

Besides input gates, a SAN model can contain *output gate* as well. An output gate has to be connected to a certain activity and to a set of places. The role of an output gate is specifying only the effect of the activity completion on the marking of the places connected to the output gate. Therefore, an output gate is characterized only by a function. The marking enabling the same activity can be expressed by means of oriented arcs, or by means of an input gate. Gates graphically appear as triangles.

In a SAN model, it is possible to set several completion cases for an activity; each case corresponds to a certain effect of the completion and has a certain probability: when the activity completes, one of the case happens. A case graphically appears as a small circle close to the activity; from the case an arc is directed to a gate or to a place. An example of SAN model is shown in Fig. 4.

The *Replicate/Join* formalism [7] was conceived for SAN models; such formalism allows to express by means of a tree structure, the way to compose together several SAN models in a unique large composed model. In the tree structure, leaf nodes are atomic SAN models, each non leaf node is a *Join* or *Replicate* operator, and the root node is the model resulting from the composition of atomic models according to the operators in the tree. In particular, the *Join* operator compose two or more SAN models by superposition over their common places; the *Replicate* operator constructs a model consisting of a number of identical copies of a certain SAN model (copies may share common places). An example of composed model is shown in Fig. 8.

The design, the composition and the analysis or simulation of SAN models is supported by the *Möbius* tool [6, 7, 10, 11, 12]. Further information about the SAN formalism can be found in [4].

# 4 Modelling the case study in form of SAN

This section presents and describes the SAN models of the case study. Each SAN represents a particular aspect, such as the control centre functions, the packet transmission, the substation functions, the intrusions, the communication network failures. The SAN models share some common places acting as the points of connection for the composition of the the models of the scenarios (see Sec. 4.4).

## 4.1 Modelling the command and signal sessions

In this section, we first consider the models about the command sessions and the signal sessions (see Sec. 2.1). They involve the control centre functions (generation of commands and polls, collection of acknowledgments and signals), the transmission of packets (commands copies, acknowledgments, poll requests, signals) by the communication network $NET1$ or $NET2$, and the substation functions (execution of commands, generation of acknowledgments, generation of signals).

### 4.1.1 Modelling the control centre functions

The functions of the control centre are the generation of commands and the collection of acknowledgments, or the generation of polls and the collection of signals. Such functions are represented by the SAN model called *"Control_Centre_Functions"* and appearing in Fig. 3 where the upper part of the model concerns the command session (command generation and the acknowledgments collection): the stochastic activity called *com_gener* models the generation of a command; the effect of its completion (defined inside the input gate *I_com_gener*) is opening the command session by marking the place *com_session_open* with one token. Moreover, such activity marks both the place *com_queue* and the place *pending_com* with 10 tokens (10 substations are present in the case study).

The generated command has to be sent to all the substations: each token inside the place *com_queue* represents a copy of the command to be transmitted to a particular substation. The tokens inside the place *com_queue* are consumed by an activity in the SAN model called *"Packets_Transmission"* in Fig. 5, where such place is present as well. The SAN model in Fig. 5 represents the transmission of packets (commands, acknowledgments, polls, signals), as described in Sec. 4.1.3.

After the generation of a command, the control centre collects the acknowledgments about the command execution, coming from the substations. The marking of the place *pending_com* corresponds to the number of command copies for which the acknowledgment has not arrived yet: the marking of the place *ack* corresponds to the incoming acknowledgments during a command session; such place is marked by an activity in the SAN model *"Packets_Transmission"* (Fig. 5). As soon as a token appears in the place *ack*, the activity *new_ack* removes the token from both the place *ack* and the place *pending_com*, according to the output gate

*O_new_ack*. In this way, we model that the control centre is aware that the command has been executed by one of the substations.

The expiration of the time out for the acknowledgments collection is represented by the deterministic activity *ack_time_out* enabled by the marking of the place *com_session_open* (input gate *I_ack_time_out*): the effect of its completion is verifying that enough acknowledgments have arrived to the control centre when the time out expires; if the place *pending_com* contains more than one token (more than one acknowledgment is missing), then the command session is considered as failed and the marking of the place *com_session_failed* is increased by one. Such place counts the number of failed command sessions. After such verification, the same activity closes the command session by removing the token inside the place *com_session_open*, as defined in the input gate *I_ack_time_out*.

We suppose that at most one command session is running at a certain time, so parallel command sessions are not possible: the input gate *I_com_gener* allows the completion of the activity *com_gener* only when the place *com_session_open* is not marked (the previous session has been closed).

The lower part of the SAN model *"Control_Centre_Functions"* in Fig. 3 is specular to the upper part, but it represents the signal sessions (the generation of polls and the collection of signals). The generation of a poll is modelled by the deterministic activity *poll_gener* opening the signal session by marking the place *sig_session_open* with one token. The same activity marks both the place *poll_queue* and the place *pending_poll* with 10 tokens, where 10 is the number of substations. Such effect of the activity *poll_gener* is specified in the input gate *I_poll_gener*. The poll has to be transmitted to all the substations, so the tokens inside the place *poll_queue* represents the poll requests to be sent to the substations. Such place is present in the SAN model *"Packets_Transmission"* in Fig. 5.

After the poll generation, the control centre collects the signals coming from the substations. The marking of the place *pending_poll* indicates the number of substations that still have to send the signal during the signal session. The incoming signals are modelled by the tokens inside the place *sig* appearing in the SAN model *"Packets_Transmission"* in Fig. 5 as well. As soon as a token appears in *sig*, the activity *new_sig* completes, removing the token from both the place *sig* and the place *pending_poll*, according to the output gate *O_new_sig*. In this way, we model that the control centre has received the signal coming from one of the substations.

The expiration of the time out for the signals collection is represented by the completion of the deterministic activity *sig_time_out* enabled by the marking of the place *sig_session_open*, as specified in its input gate *I_sig_time_out*. When this activity completes, it verifies that the place *pending_poll* does not contain more than one token. If so, the signal session has failed (more than one signal is missing), and the activity *sig_time_out* increases by one the marking of the place *sig_session_failed* counting the number of failed signal sessions. This is specified in the input gate *I_sig_time_out*.

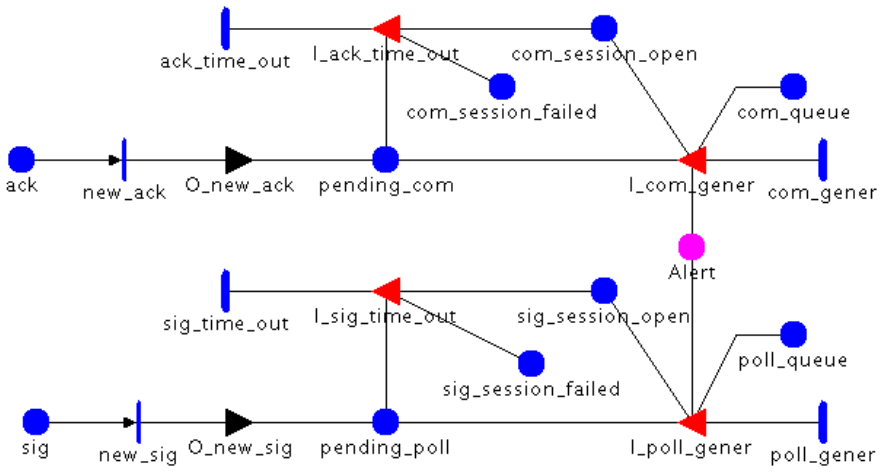


Figure 3: *"Control_Centre_Functions"*: the SAN model of the control centre functions.

We suppose that parallel signal sessions are not possible: the input gate *I_poll_gener* allows the activity *poll_gener* to complete only when the place *sig_session_open* is empty (the previous session has been closed).

Table 4 shows the completion times and the input or output gates associated with each activity in the SAN model *"Control_Centre_Functions"* (Fig. 3).

### 4.1.2 Modelling the substation functions

The functions performed by a substation are modelled in the SAN model called *"Substation_Functions"* and appearing in Fig. 4. The upper part of the model is about the execution of commands. The place *com* contains the command copies received by the substations. Such place appears also in the SAN model *"Packets_Transmission"* (Fig. 5). By means of the immediate activity *get_com* ruled by the input gate *I_get_com*, one token is moved from the place *com* into the place *ack_req* and into the place *current_com*. In this way, we model that the substation is ready to execute one of the command copies (marking of the place *ack_req*), and that no other commands will be executed during the same command session by the same substation (marking of the place *current_com*). The execution of the command and the generation of the acknowledgment are modelled by the stochastic activity *com_exec* moving the token from the place *ack_req* to the place *ack_queue* representing the presence of acknowledgments to be transmitted to the control centre. The place *ack_queue* is present also in the model *"Packets_Transmission"* (Fig. 5).

The central part of the model *"Substation_Functions"* in Fig. 4 concerns the generation of signals by the substation. The marking of the place *poll* represents the poll requests received by the substations. Such place appears also in the SAN model *"Packets_Transmission"* in Fig. 5. By means of the immediate activity *get_poll* ruled by the input gate *I_get_poll*, one token is moved from the place *poll* into both the place *sig_req*

| | |
|---|---|
| Activity: | *com_gener* (stochastic) |
| completion rate: | 1.66667E-01 $h^{-1}$ |
| input gate: | *I_com_gener* |
| input gate predicate: | (com_session_open->Mark()==0) && (Alert->Mark() > 0) |
| input gate function: | com_session_open->Mark()=1;<br>pending_com->Mark()=10;<br>com_queue->Mark()=10; |
| Activity: | *new_ack* (immediate) |
| output gate: | *O_new_ack* |
| output gate function: | if (pending_com->Mark() > 0)<br>pending_com->Mark()–; |
| Activity: | *ack_time_out* (deterministic) |
| time to complete: | 5.55556E-03 $h$ |
| input gate: | *I_ack_time_out* |
| input gate predicate: | com_session_open->Mark()==1 |
| input gate function: | com_session_open->Mark()=0;<br>if (pending_com->Mark() > 1)<br>com_session_failed->Mark()++; |
| Activity: | *poll_gener* (deterministic) |
| time to complete: | 8.33333E-02 $h$ |
| input gate: | *I_poll_gener* |
| input gate predicate: | (sig_session_open->Mark()==0) && (Alert->Mark() > 0) |
| input gate function: | sig_session_open->Mark()=1;<br>pending_poll->Mark()=10;<br>poll_queue->Mark()=10; |
| Activity: | *new_sig* (immediate) |
| output gate: | *O_new_sig* |
| output gate function: | if (pending_poll->Mark() > 0)<br>pending_poll->Mark()–; |
| Activity: | *sig_time_out* (deterministic) |
| input gate: | *I_ack_time_out* |
| input gate predicate: | sig_session_open->Mark()==1 |
| output gate function: | sig_session_open->Mark()=0;<br>if (pending_poll->Mark() > 1)<br>sig_session_failed->Mark()++; |

Table 4: The activities in the SAN model *"Control_Centre_Functions"* (Fig. 3).

and the place *current_poll*. In this way, we model that the substation is ready to generate a signal as a reply to a poll request (marking of the place *sig_req*), and that no other signals will be generated during the session by the same substation (marking of the place *current_poll*). The generation of the signal is modelled by the stochastic activity *sig_gener* moving the token from *sig_req* into the place *sig_queue* representing the signals to be transmitted to the control centre. The place *sig_queue* appears in the SAN model *"Packets_Transmission"* in Fig. 5 as well.

The lower part of the model *"Substation_Functions"* in Fig. 4 concerns the discard or the execution of fake commands received from the attacker. When the attacker generates a fake command, a copy of it is sent to each substation by means of $NET1$ or $NET2$. The marking of the place *fake* represents the fake command copies received by the substations. Such place appears also in the SAN model *"Packets_Transmission"* in Fig. 5. By means of the immediate activity *get_fake* ruled by the input gate *I_get_fake*, one token is moved from the place *fake* into both the place *fake_req* and the place *current_fake*. In this way, we model that the substation is ready to deal with the fake command by discarding or executing it (marking of the place *sig_req*), and that no other copy of the same fake command will be considered by the same substation (marking of the place *current_fake*). The discard or the execution of the fake command is modelled by the stochastic activity *fake_exec*; such activity removes the token inside the place *fake_req* and is characterized by two completion cases modelling the execution or the discard of the fake command respectively: if the first case happens, then the place *substation_ko* modelling the state of the substation becomes marked with one token, in order to represent the failed state. If instead the second case happens, such place remains empty (the substation is still available).

The functions of the substation (execution of (fake) commands and generation of signals) can not be performed if the substation is currently failed. If the place *substation_ko* becomes marked, then all the immediate activities *get_com*, *get_poll* and *get_fake* are disabled, while all the immediate activities *discard_com*, *discard_poll* and *discard_fake* are enabled according to the predicate defined in the input gates *I_discard_com*, *I_discard_poll* and *I_discard_fake* respectively. In this situation, one token in the places *com*, *poll* or *fake* is consumed if they are marked, but no (fake) commands are executed and no signals are generated. In this way, we model that in case of substation failure, though a (fake) command copy or a poll request is received by the substation, there is no reaction by the substation until its recovery: the stochastic activity *recovery* removes the token inside the place *substation_ko* in order to represent that the substation has turned back to the working state.

Table 5 and Table 6 summarize the activities inside the model *"Substation_Functions"* (Fig. 4), including the predicates and the functions of the gates ruling the completion of the activities.

| | |
|---|---|
| Activity: | *get_com* (immediate) |
| input gate: | *I_get_com* |
| input gate predicate: | substation_ko->Mark()==0 && current_com->Mark()==0 && com->Mark()>0 |
| input gate function: | current_com->Mark()=1; com->Mark()–; |
| Activity: | *com_exec* (stochastic) |
| completion rate: | 3600 $h^{-1}$ |
| Activity: | *discard_com* (immediate) |
| input gate: | *I_discard_com* |
| input gate predicate: | substation_ko->Mark()==1 && current_com->Mark()==0 && com->Mark()>0 |
| input gate function: | current_com->Mark()=1; com->Mark()–; |
| Activity: | *no_com* (immediate) |
| input gate: | *I_no_com* |
| input gate predicate: | com_session_open->Mark()==0 |
| Activity: | *get_poll* (immediate) |
| input gate: | *I_get_poll* |
| input gate predicate: | substation_ko->Mark()==0 && current_poll->Mark()==0 && poll->Mark()>0 |
| input gate function: | current_poll->Mark()=1; poll->Mark()–; |
| Activity: | *sig_gener* (stochastic) |
| completion rate: | 3600 $h^{-1}$ |
| Activity: | *discard_poll* (immediate) |
| input gate: | *I_discard_poll* |
| input gate predicate: | substation_ko->Mark()==1 && current_poll->Mark()==0 && poll->Mark()>0 |
| input gate function: | current_poll->Mark()=1; poll->Mark()–; |
| Activity: | *no_poll* (immediate) |
| input gate: | *I_no_poll* |
| input gate predicate: | sig_session_open-¿Mark()==0 |

Table 5: The activities in the SAN model *"Substation_Functions"* (Fig. 4).

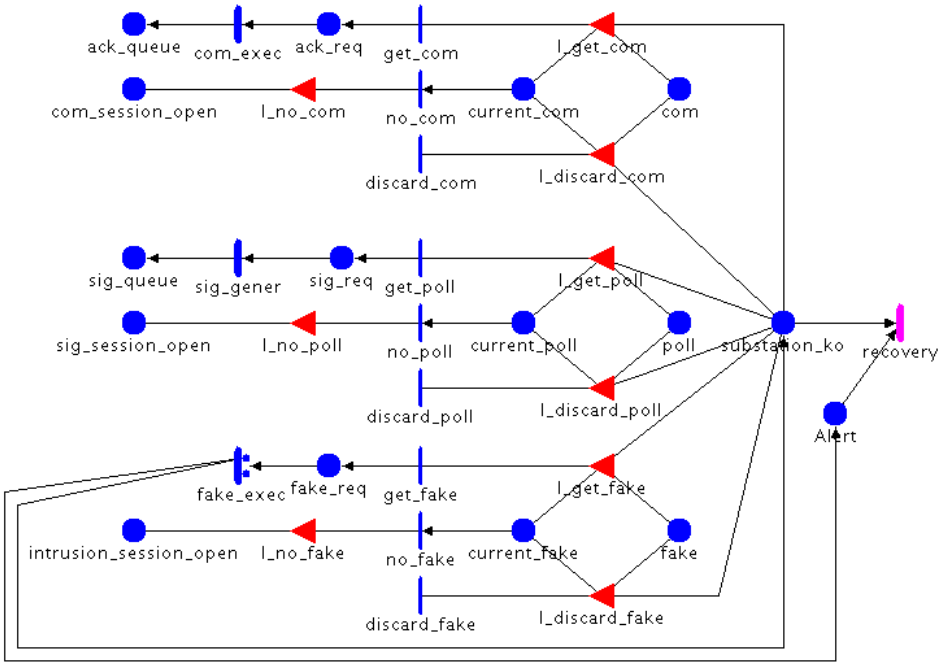


Figure 4: *"Substation_Functions"*: the SAN model of the substation functions.

| | |
|---|---|
| Activity: | *get_fake* (immediate) |
| input gate: | *I_get_fake* |
| input gate predicate: | substation_ko->Mark()==0 && current_fake->Mark()==0 && fake->Mark()>0 |
| input gate function: | current_fake->Mark()=1; fake->Mark()–; |
| Activity: | *fake_exec* (stochastic) |
| completion rate: | 3600 $h^{-1}$ |
| case 1 probability: | 0.01 |
| case 2 probability: | 0.99 |
| Activity: | *discard_fake* (immediate) |
| input gate: | *I_discard_fake* |
| input gate predicate: | substation_ko->Mark()==1 && current_fake->Mark()==0 && fake->Mark()>0 |
| input gate function: | current_fake->Mark()=1; fake->Mark()–; |
| Activity: | *no_fake* (immediate) |
| input gate: | *I_no_fake* |
| input gate predicate: | intrusion_session_open->Mark()==0 |
| Activity: | *recovery* (stochastic) |
| completion rate: | 8.33333E-02 $h^{-1}$ |

Table 6: The activities in the SAN model *"Substation_Functions"* (Fig. 4).

### 4.1.3 Modelling the packets transmission

The transmission of packets can be performed by the communication network $NET1$ or by $NET2$; packets can be command copies, acknowledgments, poll requests, signals or fake command copies. The SAN model *"Packets_Transmission"* in Fig. 5 represents this situation. The marking of the places *com_queue*, *poll_queue* and *fake_queue* represent the command copies, the poll requests and the fake command copies respectively, waiting to be transmitted on the available communication network. Such places appear also in the SAN model *"Control_Centre_Functions"* in Fig. 3. The marking of the places *ack_queue* and *sig_queue* represent acknowledgments and signals respectively, waiting to be transmitted. Such places appear also in the SAN model *"Substation_Functions"* in Fig. 4.

We suppose that the bandwidth of each communication network is equal to 16 *kbps* and that the transmission of each packet requires to consume 1 *kbps* of the bandwidth (Sec. 2.1.1). This means that no more than 16 packets can be transmitted on the same communication network at the same time. The marking of the places *com_out_1*, *poll_out_1*, *ack_out_1*, *sig_out_1* and *fake_out_1* represent the number of command copies, poll requests, acknowledgments, signals and fake command copies respectively that are currently under transmission by $NET1$. When a token appears in *com_queue*, the immediate activity *send_com* completes, removing the token, and the output gate *O_send_com* checks if the sum of the markings of *com_out_1*, *poll_out_1*, *ack_out_1*, *sig_out_1* and *fake_out_1* is less than 16 ($16kbps$ is the bandwidth of $NET1$), and if the place *net1_ko* is not marked ($NET1$ is not failed). If so, enough bandwidth is available to transmit the command copy, and the marking of the place *com_out_1* will be increased by one. If instead the sum of the markings is equal to 16 or the place *net1_ko* is marked, then no bandwidth is currently available on $NET1$: the output gate *O_send_com* will check if the sum of the markings of *com_out_2*, *poll_out_2*, *ack_out_2*, *sig_out_2* and *fake_out_2* is less than 16, and if the place *net2_ko* is not marked, in order to verify if some bandwidth is available on the communication network $NET2$. If so, the marking of *com_out_2* will be increased by one. If no bandwidth is available on both $NET1$ and $NET2$, the command copy will not be transmitted (it becomes lost). The places *net1_ko* and *net2_ko* are present also in the SAN model *"Network_Failure_and_Repair"* (Fig. 7).

The direction of poll requests, signals and fake command copies towards $NET1$ or $NET2$ is modelled in a way similar to the command copies direction: the output gates *O_send_poll*, *O_send_ack*, *O_send_sig*, *O_send_fake* perform the same checks and have the same effect of *O_send_com*, in case of completion of the activities *send_poll*, *send_ack*, *send_sig*, *send_fake* respectively, due to the presence of a token inside the places *poll_queue*, *ack_queue*, *sig_queue* *fake_queue* respectively.

The transmission of the packets by $NET1$ is modelled by the stochastic activity *transmit_1* whose completion is ruled by the input gate *I_transmit_1* having the following effect:

- any token inside *com_out_1* is moved into the place *com* which represents the command copies received

by the substations; *com* is the same place present in the SAN model *"Substation_Functions"* (Fig. 4).

- Any token inside *poll_out_1* is moved into the place *poll* which represents the poll requests received by the substations; *poll* is the same place present in the SAN model *"Substation_Functions"* (Fig. 4).
- Any token inside *ack_out_1* is moved into the place *ack* which represents the acknowledgments received by the control centre; *ack* is the same place present in the SAN model *"Control_Centre_Functions"* (Fig. 3).
- Any token inside *sig_out_1* is moved into the place *sig* which represents the signals received by the control centre; *sig* is the same place present in the SAN model *"Control_Centre_Functions"* (Fig. 3).
- Any token inside *fake_out_1* is moved into the place *fake* which represents the fake command copies received by the substations; *fake* is the same place present in the SAN model *"Substation_Functions"* (Fig. 3).

The transmission of packets by $NET2$ is modelled in a similar way by the stochastic activity *transmit_2* and the input gate *I_transmit_2* having effect on the places *com_out_2* and *com* (transmission of command copies), *poll_out_2* and *poll* (transmission of poll requests), *ack_out_2* and *ack* (transmission of acknowledgments), *sig_out_2* and *sig* (transmission of signals), *fake_out_2* and *fake* (transmission of fake command copies).

The activities present in the model *"Packets_Transmission"* in Fig. 5, together with the corresponding input or output gates, are detailed in Table 7, Table 8, Table 9 and Table 10.

## 4.2 Modelling the intrusion and the transmission of fake commands

The possibility of intrusions into the communication between the control centre and the substations, with the generation of fake commands directed to the substations, is dealt by the SAN model *"Intrusion"* in Fig. 6. The situation where no intruder is present is modelled by the presence of one token inside the place *intrusion_idle*. The occurrence of an intrusion is modelled by the activity *intrusion_begin* whose completion determines the move of the token from the place *intrusion_idle* into *intrusion_active*: this means that an intruder is present. The generation of fake commands is represented by the stochastic activity *fake_gener*: while the place *intrusion_active* is marked (input gate *I_fake_gener*), such activity is enabled to complete; each time this happens, the place *fake_queue* is marked with 10 tokens representing the fake command copies to be sent to the substations. Such tokens will be consumed by the immediate activity *fake_send* in the SAN model *"Packets_Transmission"* (Fig. 5). The completion rates about the activities in the SAN model *"Intrusion"* (Fig. 6) are reported in Table 11.

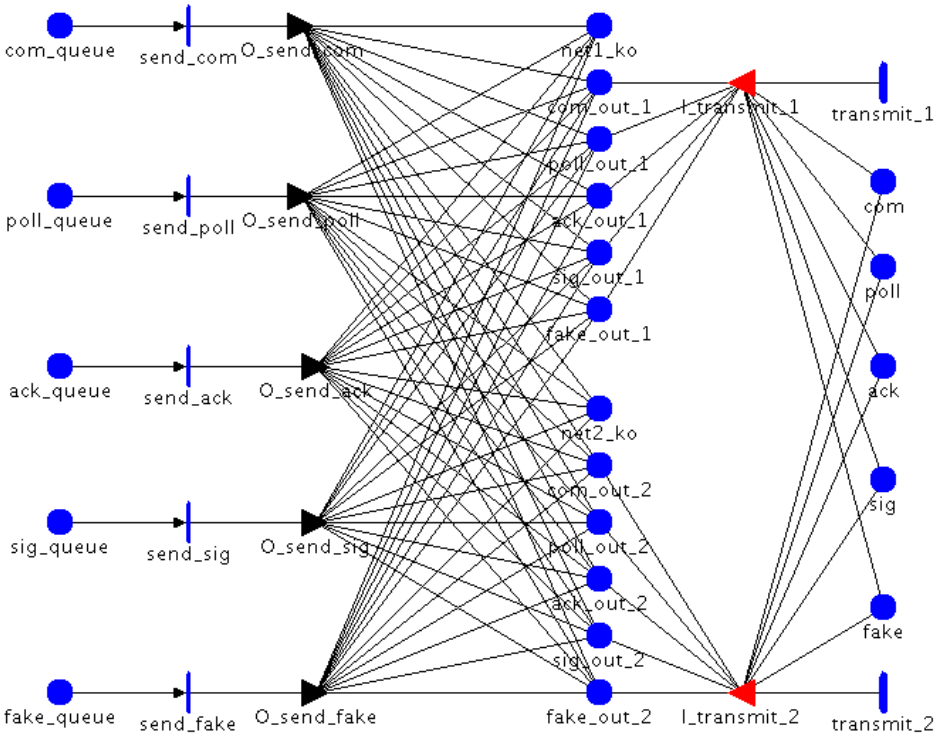


Figure 5: *"Packets_Transmission"*: the SAN model of the packet transmission.

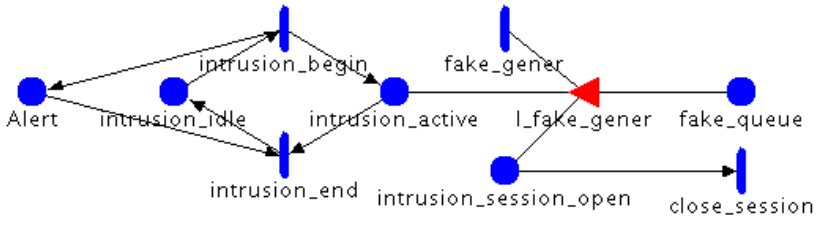


Figure 6: *"Intrusion"*: the SAN model of the intrusion and the generation of fake commands.

| | |
|---|---|
| Activity: | *send_com* (immediate) |
| output gate: | *O_send_com* |
| output gate predicate: | if ((com_out_1->Mark()+poll_out_1->Mark()+ack_out_1->Mark()+<br>sig_out_1->Mark()+fake_out_1->Mark() < 16) && (net1_ko->Mark()==0))<br>com_out_1->Mark()++;<br>else<br>if ((com_out_2->Mark()+poll_out_2->Mark()+ack_out_2->Mark()+<br>sig_out_2->Mark()+fake_out_2->Mark() ¡ 16) && (net2_ko->Mark()==0))<br>com_out_2->Mark()++; |
| Activity: | *send_poll* (immediate) |
| output gate: | *O_send_poll* |
| output gate predicate: | if ((com_out_1->Mark()+poll_out_1->Mark()+ack_out_1->Mark()+<br>sig_out_1->Mark()+fake_out_1->Mark() < 16) && (net1_ko->Mark()==0))<br>poll_out_1->Mark()++;<br>else<br>if ((com_out_2->Mark()+poll_out_2->Mark()+ack_out_2->Mark()+<br>sig_out_2->Mark()+fake_out_2->Mark() < 16) && (net2_ko->Mark()==0))<br>poll_out_2->Mark()++; |
| Activity: | *send_ack* (immediate) |
| output gate: | *O_send_ack* |
| output gate predicate: | if ((com_out_1->Mark()+poll_out_1->Mark()+ack_out_1->Mark()+<br>sig_out_1->Mark()+fake_out_1->Mark() < 16) && (net1_ko->Mark()==0))<br>ack_out_1->Mark()++;<br>else<br>if ((com_out_2->Mark()+poll_out_2->Mark()+ack_out_2->Mark()+<br>sig_out_2->Mark()+fake_out_2->Mark() < 16) && (net2_ko->Mark()==0))<br>ack_out_2->Mark()++; |

Table 7: The activities in the SAN model *"Packets_Transmission"* (Fig. 5).

| | |
|---|---|
| Activity: | *send_sig* (immediate) |
| output gate: | *O_send_sig* |
| output gate predicate: | if ((com_out_1->Mark()+poll_out_1->Mark()+ack_out_1->Mark()+<br>sig_out_1->Mark()+fake_out_1->Mark()<16) && (net1_ko->Mark()==0))<br>sig_out_1->Mark()++;<br>else<br>if ((com_out_2->Mark()+poll_out_2->Mark()+ack_out_2->Mark()+<br>sig_out_2->Mark()+fake_out_2->Mark()<16) && (net2_ko->Mark()==0))<br>sig_out_2->Mark()++; |
| Activity: | *send_fake* (immediate) |
| output gate: | *O_send_fake* |
| output gate predicate: | if ((com_out_1->Mark()+poll_out_1->Mark()+ack_out_1->Mark()+<br>fake_out_1->Mark()+fake_out_1->Mark()<16) && (net1_ko->Mark()==0))<br>fake_out_1->Mark()++;<br>else<br>if ((com_out_2->Mark()+poll_out_2->Mark()+ack_out_2->Mark()+<br>fake_out_2->Mark()+fake_out_2->Mark()<16) && (net2_ko->Mark()==0))<br>fake_out_2->Mark()++; |

Table 8: The activities in the SAN model *"Packets_Transmission"* (Fig. 5).

| | |
|---|---|
| Activity: | *transmit_1* (stochastic) |
| completion rate: | 3600 $h^{-1}$ |
| input gate: | *I_transmit_1* |
| input gate predicate: | (com_out_1->Mark()>0) \|\| (poll_out_1->Mark()>0) \|\| (ack_out_1->Mark()>0) \|\| (sig_out_1->Mark()>0) \|\| (fake_out_1->Mark()>0) |
| input gate function: | if (com_out_1->Mark() > 0) { com->Mark() = com->Mark() + com_out_1->Mark(); com_out_1->Mark() = 0; } if (poll_out_1->Mark() > 0) { poll->Mark() = poll->Mark() + poll_out_1->Mark(); poll_out_1->Mark() = 0; } if (ack_out_1->Mark() > 0) { ack->Mark() = ack->Mark() + ack_out_1->Mark(); ack_out_1->Mark() = 0; } if (sig_out_1->Mark() > 0) { sig->Mark() = sig->Mark() + sig_out_1->Mark(); sig_out_1->Mark() = 0; } if (fake_out_1->Mark() > 0) { fake->Mark() = fake->Mark() + fake_out_1->Mark(); fake_out_1->Mark() = 0; } |

Table 9: The activities in the SAN model *"Packets_Transmission"* (Fig. 5).

| | |
|---|---|
| Activity: | *transmit_2* (stochastic) |
| completion rate: | 3600 $h^{-1}$ |
| input gate: | *I_transmit_2* |
| input gate predicate: | (com_out_2->Mark()>0) \|\| (poll_out_2->Mark()>0) \|\| (ack_out_2->Mark()>0) \|\| (sig_out_2->Mark()>0) \|\| (fake_out_2->Mark()>0) |
| input gate function: | if (com_out_2->Mark() > 0) { com->Mark() = com->Mark() + com_out_2->Mark(); com_out_2->Mark() = 0; } if (poll_out_2->Mark() > 0) { poll->Mark() = poll->Mark() + poll_out_2->Mark(); poll_out_2->Mark() = 0; } if (ack_out_2->Mark() > 0) { ack->Mark() = ack->Mark() + ack_out_2->Mark(); ack_out_2->Mark() = 0; } if (sig_out_2->Mark() > 0) { sig->Mark() = sig->Mark() + sig_out_2->Mark(); sig_out_2->Mark() = 0; } if (fake_out_2->Mark() > 0) { fake->Mark() = fake->Mark() + fake_out_2->Mark(); fake_out_2->Mark() = 0; } |

Table 10: The activities in the SAN model *"Packets_Transmission"* (Fig. 5).

| | |
|---|---|
| Activity: | *intrusion_begin* (stochastic) |
| completion rate: | 1.38889E-03 $h^{-1}$ |
| Activity: | *fake_gener* (stochastic) |
| completion rate: | 1.0 $h^{-1}$ |
| input gate: | *I_fake_gener* |
| input gate predicate: | intrusion_active->Mark()==1 && intrusion_session_open->Mark()==0 |
| input gate function: | intrusion_session_open->Mark()=1; |
| | fake_queue->Mark()=10; |
| Activity: | *close_session* |
| completion time: | 5.55556E-03 $h$ |

Table 11: The activities in the SAN model *"Intrusion"* (Fig. 6).

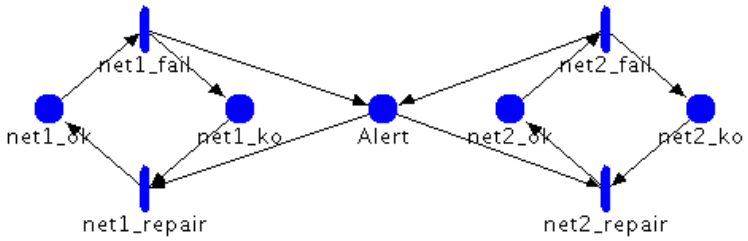


Figure 7: *"Network_Failure_and_Repair"*: the SAN model of the communication networks failure and repair.

## 4.3 Modelling the communication network failure

The failure and the repair of the communication network $NET1$ or $NET2$ is represented by the SAN model *"Network_Failure_and_Repair"* in Fig. 7. The working state of $NET1$ is represented by the presence of one token inside the place *net1_ok*. The failure of $NET1$ is modelled by the stochastic activity *net1_fail* whose effect is moving the token from the place *net1_ok* into *net1_ko*. In this way, we model the failed state of $NET1$. Its repair is represented by the stochastic activity *net1_repair* whose completion determines the move of the token from the place *net1_ko* into *net1_ok*. In this way, we model the return of $NET1$ to the working state.

The failure and repair of $NET2$ is modelled in a specular way in the right side of the SAN model *"Network_Failure_and_Repair"* in Fig. 7. The places *net1_ko* and *net2_ko* appear also in the SAN model *"Packets_Transmission"* (Fig. 5). The completion rates of the activities in the model *"Network_Failure_and_Repair"* (Fig. 7) are reported in Table 12.

## 4.4 Modelling the scenarios

In the previous sections, we presented the SAN model for each aspect of the case study. Several SAN models have some common places, for instance the place *net1_ko* is present both in *"Packets_Transmission"* and in *"Network_Failure_and_Repair"* ((Fig. 5) and Fig. 7) modelling the transmission of packets and the

| | |
|---|---|
| Activity: | *net1_fail* (stochastic) |
| completion rate: | 1.38889E-03 $h^{-1}$ |
| Activity: | *net1_repair* (stochastic) |
| completion rate: | 8.33333E-02 $h^{-1}$ |
| Activity: | *net2_fail* (stochastic) |
| completion rate: | 1.38889E-03 $h^{-1}$ |
| Activity: | *net2_repair* (stochastic) |
| completion rate: | 8.33333E-02 $h^{-1}$ |

Table 12: The activities in the SAN model *"Network_Failure_and_Repair"* (Fig. 7).

communication network failure and repair, respectively. The *Möbius* tool allows to merge together the several SAN model by superposition over the common places. Moreover, it allows to replicate several times the same SAN model, with the possibility for the model replicas, of sharing some places. To this aim, it is possible to build a composed model based on the *Join* and the *Replicate* operators allowing the merge and the replication of SAN models respectively, as mentioned in Sec. 3.

In this report, we are interested to evaluate the case study in the scenarios defined in Sec. 2.3. The model of each scenario is obtained by replicating and merging some of the SAN models presented in the previous sections. The composed model for the Scenario 1 is shown in Fig. 8 where the SAN model *"Substation_Functions"* is replicated 10 times in order to model the presence of 10 substations: the places *com*, *ack_queue*, *poll*, *sig_queue* and *fake* are shared by the replicas. The result of the replication, *"Substation_Set"*, is merged (joined) with the SAN models *"Control_Centre_Functions"* and *"Packets_Transmission"* concerning the command and signal sessions, and with the SAN model *"Intrusion"* concerning the possibility of intrusions and generation of fake commands.

The composed model for the Scenario 2 is obtained by composing the SAN models according to Fig. 9 where *"Substation_Functions"* is still replicated 10 times in order to represent the presence of 10 substations. The result, *"Substation_Set"*, is joined with *"Control_Centre_Functions"* and *"Packets_Transmission"* in order to model the command and signal sessions, and with the SAN model *"Network_Failure_and_Repair"* about the failure and the repair of the communication networks.

The composed model for the Scenario 3 is shown in Fig. 10 where *"Substation_Functions"* is replicated 10 times in order to obtain the set of substations (*"Substation_Set"*) which is in turn joined with *"Control_Centre_Functions"*, *"Packets_Transmission"*, *"Network_Failure_and_Repair"*, and *"Intrusion"*. In this way both the intrusions and the communication network failures are considered in the scenario.

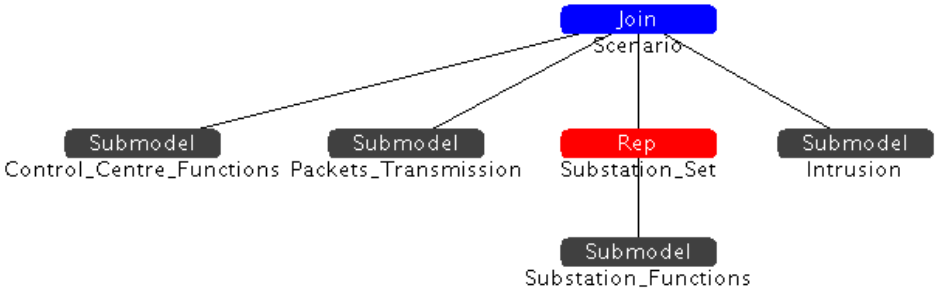


Figure 8: The composed model of the Scenario 1.

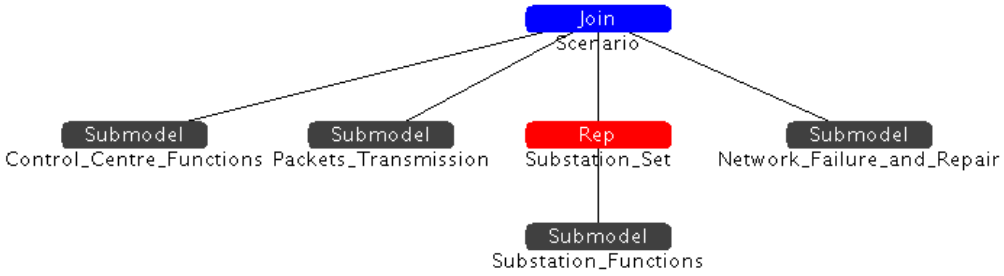


Figure 9: The composed model of the Scenario 2.

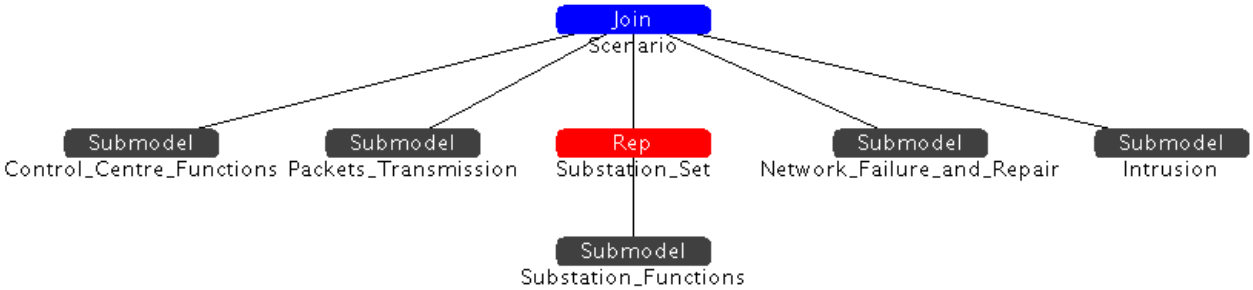


Figure 10: The composed model of the Scenario 3.

# 5 Simulation results

The composed models described in the previous section have been object of simulation. For each scenario, 10000 simulation batches have been performed by means of *Möbius*, requiring a confidence level equal to 0.95, and a relative confidence interval equal to 0.1. The measures computed by the simulation are :

- $Pr_{com}(t)$: the probability that at least one command session has failed at a certain time;
- $Pr_{sig}(t)$: the probability that at least one signal session has failed at a certain time;
- $Num_{com}(t)$: mean number of failed command sessions at a certain time;
- $Num_{sig}(t)$: mean number of failed command sessions at a certain time.

All measures are computed for a mission time varying between 0 and 10000 $h$.

The first measure, $Pr_{com}(t)$, is computed as the mean value over the 10000 simulation batches, of the reward $rew1$ having the following expression:

```
if (Control_Centre_Functions->com_session_failed->Mark()>0)
  rew1=1;
else
  rew1=0;
```

This means that in each simulation batch and at a certain time, $rew1$ is equal to 1 if the place *com_session_failed* contains at least one token, or it is equal to 0 if the same place is empty. The place *com_session_failed* is present in the SAN model *"Control_Centre_Functions"* (Fig. 3) and it indicates the number of failed command sessions. So, the mean value of $rew1$ at a certain time, over the 10000 simulation batches, provides the probability that at least one command session has failed at a certain time. The values of $Pr_{com}(t)$ returned by the simulation for each scenario are reported in Table 13 and are depicted in Fig. 11.

The measure $Pr_{sig}(t)$ is computed in a similar way: it is the mean value over the 10000 simulation batches, of the reward $rew2$ whose expression follows:

```
if (Control_Centre_Functions->sig_session_failed->Mark()>0)
  rew2=1;
else
  rew2=0;
```

The place *sig_session_failed* is present in Fig. 3 and it indicates the number of failed signal sessions. The mean value of $rew2$ as a function of the time, provides the value of $Pr_{sig}(t)$. Table 14 shows the results obtained for such measure in each scenario. The same results appear in Fig. 12.

Both Fig. 11 and Fig. 12 show that according to the event occurrence times specified in Sec. 2 and the SAN models described in Sec. 4, the intrusions (Scenario 1) determine a higher probability of command or

| time | $Pr_{com}(t)$ in Scenario 1 | $Pr_{com}(t)$ in Scenario 2 | $Pr_{com}(t)$ in Scenario 3 |
|---|---|---|---|
| 1000 $h$ | 6,960E-02 | 4,880E-02 | 1,147E-01 |
| 2000 $h$ | 1,402E-01 | 9,110E-02 | 2,206E-01 |
| 3000 $h$ | 2,031E-01 | 1,366E-01 | 3,111E-01 |
| 4000 $h$ | 2,627E-01 | 1,768E-01 | 4,012E-01 |
| 5000 $h$ | 3,161E-01 | 2,180E-01 | 4,723E-01 |
| 6000 $h$ | 3,688E-01 | 2,565E-01 | 5,400E-01 |
| 7000 $h$ | 4,140E-01 | 2,926E-01 | 5,944E-01 |
| 8000 $h$ | 4,576E-01 | 3,258E-01 | 6,422E-01 |
| 9000 $h$ | 4,963E-01 | 3,608E-01 | 6,835E-01 |
| 10000 $h$ | 5,322E-01 | 3,913E-01 | 7,178E-01 |

Table 13: The probability that at least one command session has failed ($Pr_{com}(t)$) in Scenario 1, Scenario 2, Scenario 3 (the same values are reported in Fig. 11).

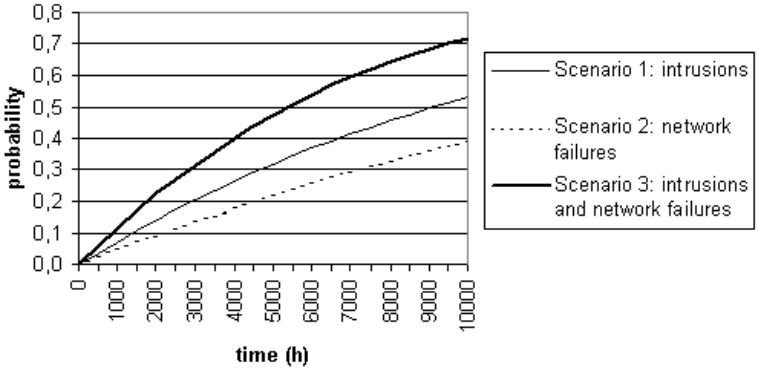


Figure 11: The probability that at least one command session has failed ($Pr_{com}(t)$) in Scenario 1, Scenario 2, Scenario 3 (the same values are reported in Tab. 13).

| time | $Pr_{sig}(t)$ in Scenario 1 | $Pr_{sig}(t)$ in Scenario 2 | $Pr_{sig}(t)$ in Scenario 3 |
|---|---|---|---|
| 1000 $h$ | 1,162E-01 | 7,060E-02 | 1,746E-01 |
| 2000 $h$ | 2,261E-01 | 1,305E-01 | 3,259E-01 |
| 3000 $h$ | 3,192E-01 | 1,888E-01 | 4,480E-01 |
| 4000 $h$ | 4,035E-01 | 2,468E-01 | 5,521E-01 |
| 5000 $h$ | 4,747E-01 | 3,024E-01 | 6,348E-01 |
| 6000 $h$ | 5,375E-01 | 3,519E-01 | 7,041E-01 |
| 7000 $h$ | 5,937E-01 | 3,962E-01 | 7,592E-01 |
| 8000 $h$ | 6,446E-01 | 4,348E-01 | 8,022E-01 |
| 9000 $h$ | 6,871E-01 | 4,749E-01 | 8,389E-01 |
| 10000 $h$ | 7,261E-01 | 5,080E-01 | 8,677E-01 |

Table 14: The probability that at least one signal session has failed ($Pr_{sig}(t)$) in Scenario 1, Scenario 2, Scenario 3 (the same values are reported in Fig. 12).

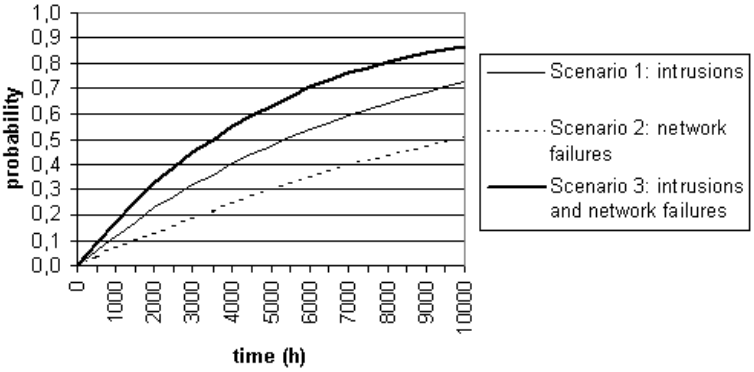


Figure 12: The probability that at least one signal session has failed ($Pr_{sig}(t)$) in Scenario 1, Scenario 2, Scenario 3 (the same values are reported in Tab. 14).

signal session failure, with respect to the communication network failures (Scenario 2). In the Scenario 1, a command or signal session fails if at least two substations are unavailable at the same time due to execution of fake commands ($N-1$ acknowledgments (signals) have to be received by the control centre before the closure of the command (signal) session). In the Scenario 2 instead, some packets are lost causing the session failure, if a communication network ($NET1$ or $NET2$) is failed and a command session is running in parallel with a signal session, or if an intruder is generating a fake command in parallel with a command (signal) session. In this situation, it may happen that the available communication network is not enough to transmit all the packets (command copies, acknowledgments, poll requests, signals, fake command copies). Still in the Scenario 2, if both communication networks are failed at the same time, this will lead to the loss of packets. Given the simulation results obtained for the Scenario 1 and the Scenario 2, we can conclude that the intrusions have a higher negative influence to the communication reliability, with respect to the network failures.

In the Scenario 3, a command or a signal session can fail due to an intrusions, to the substations unavailability as a consequence of the execution of a fake command, or to both causes. For instance, a command session may fail because one acknowledgment is missing due to the current unavailability of one substation, and another acknowledgment is missing due to a failure affecting one of the communication networks. Actually, observing both Fig. 11 ($Pr_{com}(t)$) and Fig. 12, ($Pr_{sig}(t)$) and the corresponding values in Table 13 and in Table 14 respectively, we notice that the probability of failure of a command (signal) session at a certain time in the Scenario 3, is always higher than the corresponding probabilities in the Scenario 1 and in the Scenario 2. This is due to the occurrence of both causes of command or signal session failure.

The values obtained for $Pr_{com}(t)$ and $Pr_{sig}(t)$ are compared in Fig. 13 (Scenario 1), in Fig. 14 (Scenario 2) and in Fig. 15 (Scenario 3). In all the scenarios, we can notice that $Pr_{sig}(t)$ is higher than $Pr_{com}(t)$ for any

| time | $Num_{com}(t)$ in Scenario 1 | $Num_{com}(t)$ in Scenario 2 | $Num_{com}(t)$ in Scenario 3 |
|---|---|---|---|
| 1000 $h$ | 1,6190E-01 | 7,6400E-02 | 2,3490E-01 |
| 2000 $h$ | 3,3100E-01 | 1,4070E-01 | 4,9440E-01 |
| 3000 $h$ | 4,9900E-01 | 2,1330E-01 | 7,3210E-01 |
| 4000 $h$ | 6,7590E-01 | 2,8110E-01 | 9,8530E-01 |
| 5000 $h$ | 8,5270E-01 | 3,6060E-01 | 1,2226E+00 |
| 6000 $h$ | 1,0354E+00 | 4,3330E-01 | 1,4790E+00 |
| 7000 $h$ | 1,2031E+00 | 5,0790E-01 | 1,7199E+00 |
| 8000 $h$ | 1,3759E+00 | 5,7990E-01 | 1,9613E+00 |
| 9000 $h$ | 1,5411E+00 | 6,5740E-01 | 2,2088E+00 |
| 10000 $h$ | 1,7088E+00 | 7,2770E-01 | 2,4378E+00 |

Table 15: The mean number of failed signal sessions ($Num_{com}(t)$) in Scenario 1, Scenario 2, Scenario 3 (the same values are reported in Fig. 16).

time; this is due to the fact that during a situation leading to the loss of packets (substations unavailability, communication network failure, or both), the number of signal sessions performed is higher than the number of command sessions (the time between two signal sessions is 5 $min.$, while the time between two command sessions is 6 $h$ on average). This leads to a higher number of failed signal sessions, with respect to the number of failed command sessions, as confirmed by the results obtained for the next measures.

The mean number of failed command sessions ($Num_{com}(t)$) is computed as the mean value over the 10000 simulation batches, of the reward $rew3$ whose expression is:

```
rew3=Control_Centre_Functions->com_session_failed->Mark();
```

This means that $rew3$ is equal to the marking of the place *com_session_failed* in the model *"Control_Centre_Functions"* (Fig. 3); therefore $rew3$ in a certain batch and at a certain time is equal to the number of failed command sessions at that time. The mean value of $rew3$ at a certain time, over the 10000 simulation batches, provides the mean value of failed command sessions at that time ($Num_{com}(t)$). The values of $Num_{com}(t)$ returned by the simulation for each scenario are reported in Table 15 and are depicted in Fig. 16 (consider that the number of command sessions during 10000 $h$ is about 1665).

The measure $Num_{sig}(t)$ is computed in a similar way: it corresponds to the mean value of the reward $rew4$ equal to the marking of the place *sig_session_failed* in Fig. 3:

```
rew4=Control_Centre_Functions->sig_session_failed->Mark();
```

The value of $rew4$ in a certain batch and at a certain time provides the number of failed signal sessions. The mean value of $rew4$ at a certain time, over the 10000 simulation batches, provides $Num_{sig}(t)$ at the same time. Table 16 shows the results obtained for such measure in each scenario. The same results appear in Fig. 17 (consider that the number of signal sessions during 10000 $h$ is 112500).

The results obtained for the measures $Num_{com}(t)$ (Fig. 16) and $Num_{sig}(t)$ (Fig. 17) confirm that the intrusions (Scenario 1) negatively influence the communication reliability, more than the network failures do

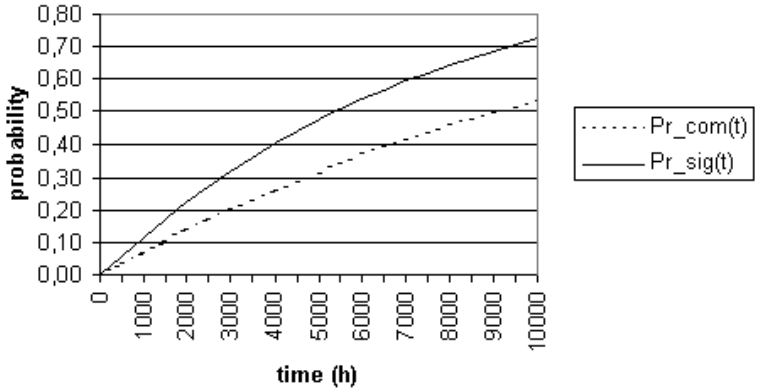


Figure 13: Scenario 1: the probability that at least one command session has failed ($Pr_{com}(t)$); the probability that at least one command session has failed ($Pr_{sig}(t)$).

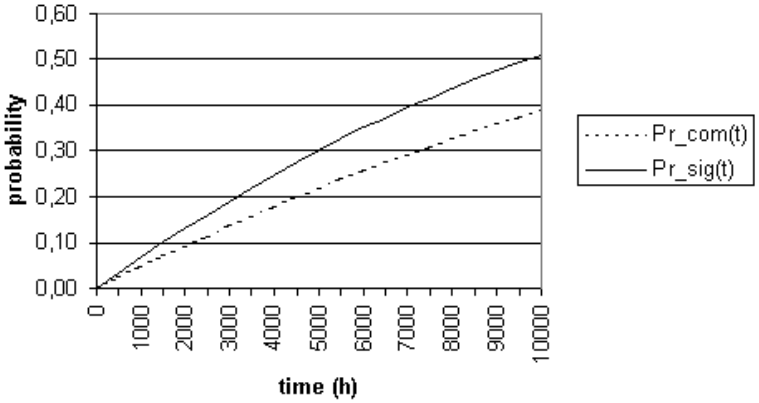


Figure 14: Scenario 2: the probability that at least one command session has failed ($Pr_{com}(t)$); the probability that at least one command session has failed ($Pr_{sig}(t)$).

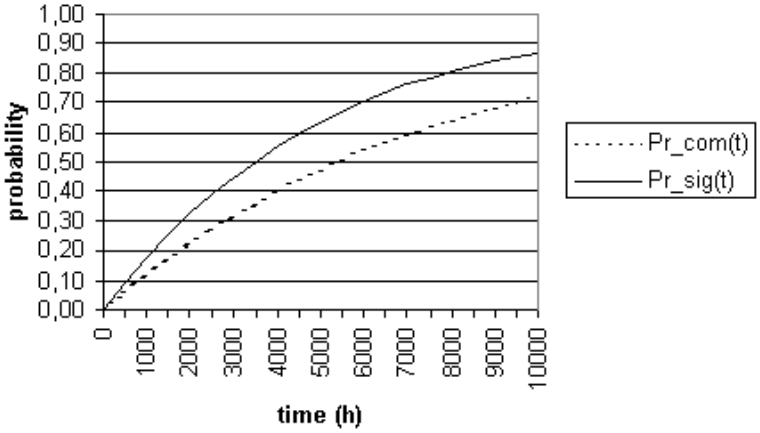


Figure 15: Scenario 3: the probability that at least one command session has failed ($Pr_{com}(t)$); the probability that at least one command session has failed ($Pr_{sig}(t)$).

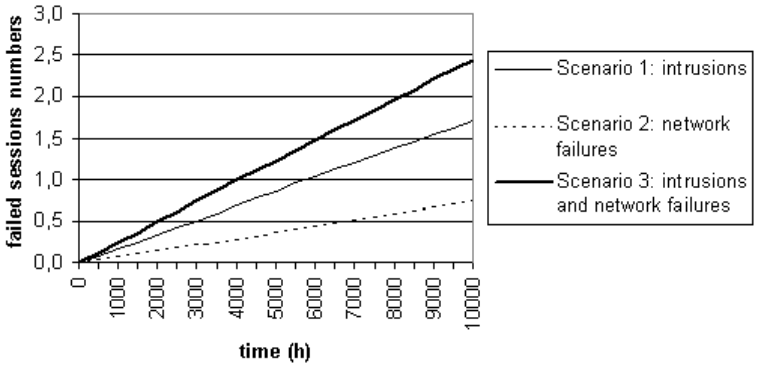


Figure 16: The mean number of failed command sessions ($Num_{com}(t)$) in Scenario 1, Scenario 2, Scenario 3 (the same values are reported in Tab. 15).

| time | $Num_{sig}(t)$ in Scenario 1 | $Num_{sig}(t)$ in Scenario 2 | $Num_{sig}(t)$ in Scenario 3 |
|---|---|---|---|
| 1000 $h$ | 1,124140E+01 | 3,203100E+00 | 1,379090E+01 |
| 2000 $h$ | 2,242070E+01 | 5,795200E+00 | 2,845390E+01 |
| 3000 $h$ | 3,380230E+01 | 8,690400E+00 | 4,258610E+01 |
| 4000 $h$ | 4,550530E+01 | 1,155400E+01 | 5,777010E+01 |
| 5000 $h$ | 5,712560E+01 | 1,484340E+01 | 7,253720E+01 |
| 6000 $h$ | 6,902630E+01 | 1,781620E+01 | 8,751700E+01 |
| 7000 $h$ | 8,086260E+01 | 2,073570E+01 | 1,017083E+02 |
| 8000 $h$ | 9,275740E+01 | 2,389450E+01 | 1,161295E+02 |
| 9000 $h$ | 1,042070E+02 | 2,697450E+01 | 1,310697E+02 |
| 10000 $h$ | 1,154704E+02 | 2,978360E+01 | 1,451636E+02 |

Table 16: The mean number of failed command sessions ($Num_{sig}(t)$) in Scenario 1, Scenario 2, Scenario 3 (the same valued are reported in Fig. 17).

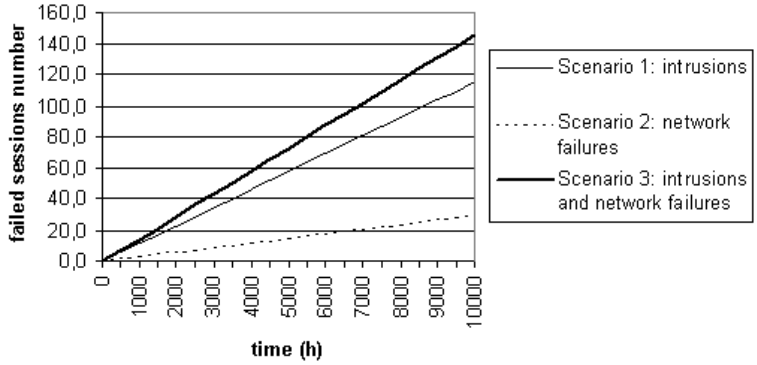


Figure 17: The mean number of failed signal sessions ($Num_{sig}(t)$) in Scenario 1, Scenario 2, Scenario 3 (the same valued are reported in Tab. 16).

(Scenario 2). In Scenario 3, the occurrence of both causes of command or signal session failure, determine higher values for $Num_{com}(t)$ and $Num_{sig}(t)$, with respect to the Scenario 1 and the Scenario 2, as it happens for $Pr_{com}(t)$ and $Pr_{sig}(t)$.

# 6 Conclusions

This report presented the modelling of communication scenarios between one control centre and a set of substations exchanging commands and signals inside an area of a distribution grid, where the communication reliability may be affected by attacks, failures, or both. The scenarios have been modelled, simulated and evaluated by means of SAN allowing to represent both the deterministic and the stochastic events, in case of normal functioning and in case of threats. The simulation of the SAN models of the scenarios provided the effect of the threats in terms of probability and quantity of lost data.

**Acknowledgments.** This work has been partially supported by the EU Project CRUTIAL IST-2004-27513.

# References

[1] D. Codetta-Raiteri and R. Nai. Evaluation of communication scenarios inside the Electrical Power System. *International Journal of Modelling and Simulation*, (to appear).

[2] CRUTIAL project's web page. http://crutial.cesiricerca.it.

[3] F. Garrone(editor), C. Brasca, D. Cerotti, D. Codetta-Raiteri, A. Daidone, G. Deconinck, S. Donatelli, G. Dondossola, F. Grandoni, M. Kaaniche, and T. Rigole. *Deliverable D2: Analysis of new control applications*. CRUTIAL project, http://crutial.cesiricerca.it/Dissemination/DELIVERABLES-OF-THE-PROJECT.asp, January 2007.

[4] W. H. Sanders and J. F. Meyer. Stochastic activity networks: Formal definitions and concepts. *Lecture Notes in Computer Science*, 2090:315–343, 2001.

[5] M. Ajmone-Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modelling with Generalized Stochastic Petri Nets*. J. Wiley and Sons, 1995.

[6] T. Courtney, D. Daly, S. Derisavi, S. Gaonkar, M. Griffith, V. Lam, and W. H. Sanders. The Möbius Modeling Environment: Recent Developments. In *Proceedings of the 1st International Conference on Quantitative Evaluation of Systems (QEST)*, pages 328–329, Twente, The Netherlands, September 2004.

[7] D. Deavours, G. Clark, T. Courtney, D. Daly, S. Derisavi, J. Doyle, W. Sanders, and P. G. Webster. The Möbius Framework and its Implementation. *IEEE Transactions on Software Engineering*, 28(10):956–969, 2002.

[8] G. Chiola, C. Dutheillet, G. Franceschinis, and S. Haddad. Stochastic Well-Formed Colored Nets and Symmetric Modeling Applications. *IEEE Transactions on Computers*, 42:1343–1360, 1993.

[9] G. Chiola, C. Dutheillet, G. Franceschinis, and S. Haddad. Stochastic Well-Formed Coloured Nets and Multiprocessor Modelling Applications. In K. Jensen and G. Rozenberg, editors, *High-Level Petri Nets. Theory and Application*. Springer Verlag, 1991.

[10] G. Clark, T. Courtney, D. Daly, D. Deavours, S. Derisavi, J. M. Doyle, W. H. Sanders, and P. Webster. The Möbius Modeling Tool. In *Proceedings of the 9th International Workshop on Petri Nets and Performance Models (PNPM)*, pages 241–250, Aachen, Germany, September 2001.

[11] T. Courtney, D. Daly, S. Derisavi, V. Lam, and W. H. Sanders. The Möbius Modeling Environment. In *Proceedings of Tools of the Illinois International Multiconference on Measurement, Modelling, and Evaluation of Computer-Communication Systems*, pages 34–37, 2003.

[12] J. M. Doyle. Abstract models specification using the Möbius modeling tool. Master's thesis, University of Illinois, 2000.