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

Simulating the communication of commands and signals in a distribution grid

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

TECHNICAL REPORT TR-INF-2009-12-08-UNIPMN
(December 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

2009-07 *A temporal relational data model for proposals and evaluations of updates*, Anselma, L., Bottrighi, A., Montani, S., Terenziani, P., September 2009.

2009-06 *Performance analysis of partially symmetric SWNs: efficiency characterization through some case studies*, Baarir, S., Beccuti, M., Dutheillet, C., Franceschinis, G., Haddad, S., July 2009.

2009-05 *SAN models of communication scenarios inside the Electrical Power System*, Codetta-Raiteri, D., Nai, R., July 2009.

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.

Contents

Simulating the communication of commands and signals in a distribution grid

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

The report presents the simulation of communication scenarios involving one area control centre and a set of substations inside a distribution grid of the Electrical Power System. In such scenarios, the communication is affected by threats different from those under exam in [1, 2]; in particular, here, we consider the denial of service attack to the communication network, and the temporary internal failure of a subset of substations. The scenarios have been modeled and simulated in form of Stochastic Activity Networks (SAN); the goal is the evaluation of the impact of the threats, on the communication reliability.

Acronym list:

DoS	Denial of Service
EPS	Electrical Power System
FT	Fault Tree
ICT	Information Communication Technology
IED	Intelligent Electronic Device
LAN	Local Area Network
MCDTU	Monitoring Control and Defense Terminal Unit
SAN	Stochastic Activity Network
SPN	Stochastic Petri Net

1 Introduction

This work was developed inside CRUTIAL project (*CRitical UTility InfrastructurAL resilience*) [3] investigating the ways to obtain the resilience of the Electrical Power System (EPS); this means the capacity of the EPS to provide its service despite of the occurrence of failures or attacks concerning devices, applications or functionalities inside the system. Actually, in the EPS, an accidental failure or a malicious attack may affect a subset of the EPS infrastructures; For instance, an attack to a communication network may affect the data information or command exchange among the EPS sites connected by that network; as a consequence, such attack may compromise an automation function depending on such data, such as the teleoperation or the

voltage regulation [4, 5]. This may cause physical damages to the infrastructures or compromise the electric power supply.

The EPS can be structured in three subsystems, each composed by a physical and an ICT infrastructure:

- the *Power generation* consists of the set of plants generating the electric power;
- the *Transmission grid* is the set of high voltage 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 or low voltage electric lines, substations and control centres in charge of transporting the electric power to the consumers located in the region.

One of the activities in CRUTIAL is the evaluation of critical scenarios. Such a scenario consists of a particular event sequences occurring in a certain portion of the EPS (scenario *domain*), as a consequence of an attack or a failure. Each scenario is characterized by the occurrence of a particular kind of these threats. One of the ways to evaluate the scenarios is the simulation of stochastic models representing the events in the scenarios; the goal is estimating the effects of attacks or failures, on the scenario domain.

The critical scenarios of interest in the project are defined in [5] and take place in different domains. In particular, several scenarios deal with the communication between the sites of the EPS (control centres, substations, plants, etc.). The communication can be compromised by attacks or failures affecting the sites or the communication networks. In [1, 2], we evaluated communication scenarios involving a control centre and a set of substations located in a distribution grid; in [1, 2], the scenarios are characterized by intrusions and communication network failures. In this report instead, the scenarios have the same domain (Sec. 2), but they are characterized by *denial of service* (DoS) attacks affecting the communication network, and the failures of the substation components (Sec. 3). The scenarios are modeled (Sec. 5) and simulated (Sec. 6) in form of *Stochastic Activity Network* (SAN) [6], by means of the *Möbius* tool [7, 8]; the goal is estimating the communication reliability in terms of probability and quantity of failures in the communication.

2 The scenarios domain

The scenarios under exam [5] take place in a domain composed by one control centre, a set of 10 substations, and two communication networks, inside a distribution grid of the EPS. 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 on the electrical lines. In the case of the distribution grid, the same command may be sent to all the substations. For instance, a command is an arming or disarming order [5]. The generation of a command by the control centre occurs as a

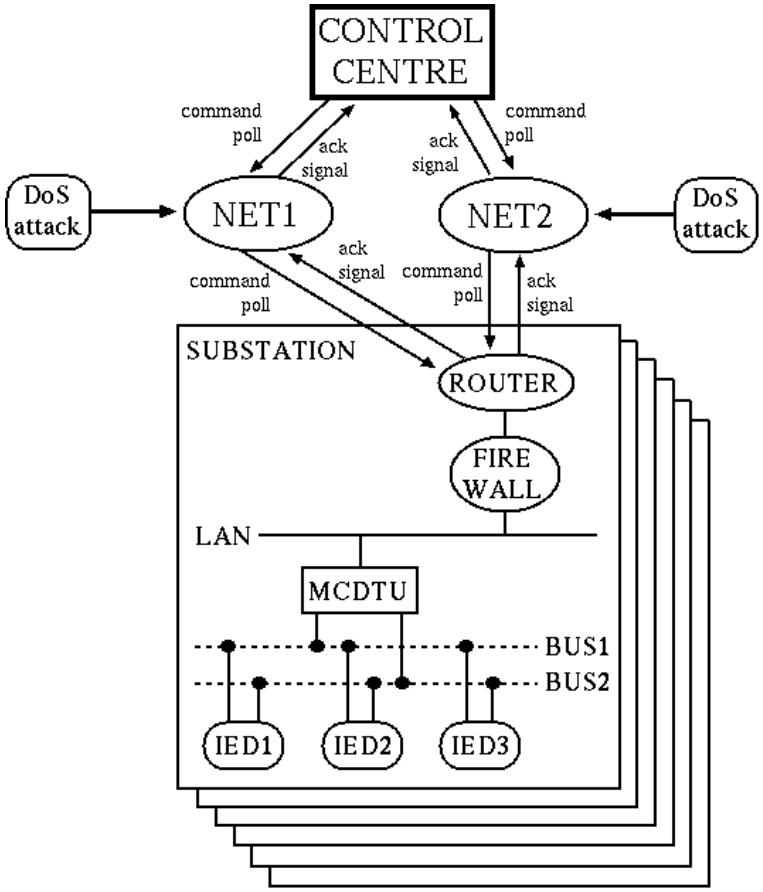

Figure 1: The scheme of the domain and the threats.

consequence of another 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. Such signals are sent periodically and describe the state of the substations or the state of the electrical lines connected to them. Such information allow the control centre to monitor the state of the portion of the distribution grid under its control. So, the communication of commands and signals has to be reliable in order to avoid malfunctioning in the distribution grid.

2.1 Command and signal sessions

In our domain, 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 a 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;
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.

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 session: 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.

We assume that that at most one command (signal) session is running at any time. In the domain under study, the time for an event to occur can be deterministic or random; in the second case, such time is ruled by the *negative exponential distribution* whose rate is the inverse value of the mean time for the event to occur. The occurrence (mean) times for the events in a command or signal session are reported in Tab. 1.

2.2 The transmission of packets

In our domain, the transmission of the several kinds of packets (command copies, acknowledgments, poll requests and signals) is performed by means of the redundant communication networks $NET1$ and $NET2$. $NET1$ is usually used for the communication between the control centre and the substations. We suppose

Event	Type of event	(mean) time to occur	occurring rate
command generation	stochastic	6.00000E+00 h	0.16667 h^{-1}
command execution	stochastic	2.77778E-04 h	3600 h^{-1}
time out for ack.	deterministic	5.55556E-03 h	-
poll generation	deterministic	8.33333E-02 h	-
signal generation	stochastic	2.77778E-04 h	3600 h^{-1}
time out for signals	deterministic	5.55556E-03 h	-
packet transmission	stochastic	2.77778E-04 h	3600 h^{-1}

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

that the bandwidth of each communication network is equal to 16 $kbit/sec$ and that the transmission of each packet consumes 1 $kbit/sec$ of the bandwidth. This means that no more than 16 packets can be transmitted on the same communication network at the same time. It may happen that the current available bandwidth of $NET1$ is not enough to transmit all the packets. For instance, if a command session and a signal session are running in parallel way, it may happen that 10 acknowledgments and 10 signals have to be transmitted to the control centre at the same time. In this case, 16 of such packets will be transmitted by $NET1$, while the remaining 4 packets will be directed to $NET2$ for the transmission.

Actually, we could have specified that the transmission of a packet requires less than 1 $kbit/sec$ of the bandwidth, or that a communication network has a bandwidth higher than 16 $kbit/sec$; 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 case study, eventually making the simulation computational costs worse.

3 The scenarios definition

In absence of attacks or failures, the communication between the control centre and the substations can not fail. In case of threats instead, some packets (command copies, acknowledgments, poll requests, signals) may be lost. If the number of substations is N, we consider a command (signal) session as successful if at least $N-1$ acknowledgments (signals) are received by the control centre before that the time out expires ($N = 10$ in the domain under study). If instead, more than one acknowledgment (signal) is missing when the time out expires, then the command (signal) session is considered to be failed.

As mentioned in Sec. 1, each scenario is characterized by the occurrence of a particular kind of attack or failure, and in this report we are interested in evaluating the domain described so far, in three scenarios:

- Scenario 1: the DoS attacks may occur;

- Scenario 2: the substations failures may occur;
- Scenario 3: both the substations failures and the DoS attacks may occur.

Such scenarios are different from those in [1, 2] where the threats under exam are the intrusion in the communication with the generation of fake commands, and the temporary unavailability of the communication network.

3.1 The DoS attack

During a DoS attack, the attacker sends a huge amount of packets on the affected communication network: the effect is the gradual reduction of the bandwidth available for normal communication, leading to the complete unavailability of the bandwidth. We assume that a DoS attack may affect $NET1$ or $NET2$; both communication networks may be attacked several times, but a communication network can not be the object of more than one attack at the same time. It may happen that both networks are under attack at the same time, but in this case, two distinct attacks are running and each affects one communication network.

$NET1$ and $NET2$ are redundant; so, in case of $NET1$ under attack, its bandwidth is gradually consumed by the packets transmitted by the attacker; therefore also $NET2$ has to be exploited to transmit. If the global available bandwidth of both $NET1$ and $NET2$ is not enough to transmit all the packets (command copies, acknowledgments, poll requests or signals), then some of them will not be transmitted becoming lost.

We assume that a DoS attack affecting a certain communication network, occurs every month on average, and that its mean duration is 12 h. Moreover, we suppose that mean time to completely consume the bandwidth of $NET1$ is 3 h: since the bandwidth of $NET1$ and $NET2$ is 16 $kbit/sec$ respectively, then the bandwidth occupancy by the DoS attack is increased by 1 $kbit/sec$ every 675 sec. (Tab. 2). When the DoS attack ends, the bandwidth consumed by the attack becomes available again for the normal communication.

Event	mean time to occur	occurring rate
DoS occurrence	720 h	0.00139 h^{-1}
DoS duration	12 h	0.08333 h^{-1}
Bandwidth reduction by 1 $kbit/sec$.	0.1875 h	5.33333 h^{-1}

Table 2: The mean occurrence time and the corresponding rates about the events in the DoS attack.

3.2 The substation failure

We assume that a substation is composed by three subsystems (Fig. 1):

- the MCDTU is the core of the substation and consists of a particular device in charge of managing the requests for command execution or for signal generation coming from the control centre. The MCDTU is connected to both the substation LAN and to the substation *bay*.

- The LAN acts as a bridge between the MCDTU and the external communication networks $NET1$ and $NET2$: all the packets transferred from the external communication networks to the LAN, then to the MCDTU (commands and polls), or in the opposite sense (acknowledgments and signals), are directed by a router and are filtered by a firewall. Actually the substation LAN could host workstations as well, but their presence is not essential to the communication scenarios considered in this report, so we avoid to consider them.

- The *bay* contains all the electrical devices necessary to physically perform the commands received by the MCDTU, and to generate the signals to be delivered to the control centre. We assume that the bay contains three redundant IED components connected to the MCDTU by means of two redundant electrical buses: the MCDTU controls the IEDs ordering them the execution of the commands or the retrieval of signals.

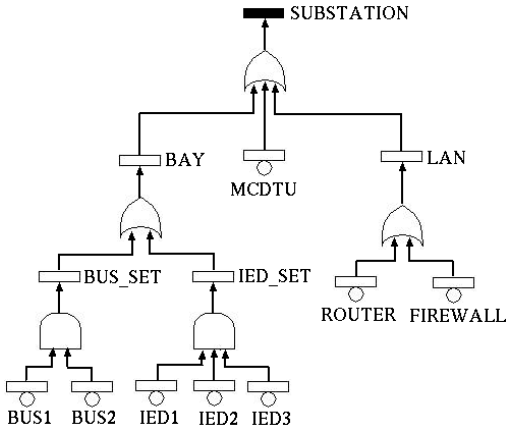

Figure 2: The Fault Tree model of the substation failure mode.

Component	MTTF	Failure Rate	MTTR	Repair Rate
bus	4380 h	2.28311E-4 h^{-1}	24 h	4.16667E-2 h^{-1}
IED	4380 h	2.28311E-4 h^{-1}	48 h	2.08333E-2 h^{-1}
MCDTU	8760 h	1.14155E-4 h^{-1}	12 h	8.33333E-2 h^{-1}
router	17520 h	5.70776E-5 h^{-1}	6 h	1.66667E-1 h^{-1}
firewall	17520 h	5.70776E-5 h^{-1}	6 h	1.66667E-1 h^{-1}

Table 3: The mean time to failure (MTTF), the failure rate, the mean time to repair (MTTR) and the repair rate of each substation component.

The failure mode of the substation can be displayed in form of *Fault Tree* (FT) [9] expressing by means of Boolean gates (AND, OR) how combinations of component failure events can lead to the failure of subsystems or of the whole system. According to the FT model in Fig. 2, the substation becomes unavailable (event

SUBSTATION) if at least one of the following events occurs: the bay fails (event BAY), the substation LAN fails (event LAN), or the MCDTU fails (event $MCDTU$). The failure of the bay (event BAY) occurs if both the bus $B1$ and the bus $B2$ are failed (event BUS_SET), or if all the IEDs are failed (event IED_SET).

While a substation is unavailable because of its internal failure, it can not execute commands or generate signals. Anyway, we assume that all the substation components are repairable, so the failure state of the substation is temporary and the substation can be available again, when the repair of failed components is completed. We suppose that each repair action concerns a single component. The mean time to failure, the mean time to repair and the corresponding rates are reported in Tab. 3.

Actually in our report, we do not resort to the *Fault Tree Analysis* [9] in the scenarios evaluation. The FT model is exploited only as a graphical representation of the failure mode of the substation, and in Sec. 4, it will be converted into the SAN model representing both the failure and the repair mode of the substation.

4 Basic notions on SAN

SAN can be considered as a particular form of Petri Net; so, a SAN model contains *places*, *activities* (transitions) and arcs. A place graphically appears as a circle, and contains a certain number of *tokens* (*marking*). A particular condition on 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.

An instantaneous activity completes (fires) as soon as it is enabled; a timed activity instead, completes after a certain amount of time. In the detailed description of the SAN models of the scenarios (Appendix A), we call "stochastic activity" a timed activity whose time to complete is a random, while we call "deterministic activity" a timed activity whose time to complete is deterministic. The condition enabling the completion of an activity can be expressed by connecting the activity to the places by means of oriented arcs, as it is possible in SPN. The effect of the activity completion on the places can be specified in the same way. Another way to express the condition enabling a certain activity consists of using *input gates*, graphically appearing as red triangles. 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 gates* as well; they appear as black triangles. An output gate has to be connected to a certain activity and to a set of standard or extended 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.

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 cases happens. A case graphically appears as a small circle close to the activity; from the case an arc is directed to an output gate or a place.

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).

4.1 Motivating the use of SAN

We have chosen SAN as modeling formalism because it inherits the modeling power of Petri Nets and introduces some advantages. As Petri Net based formalisms in general, SAN allows to express the system states and behavior in terms of places containing tokens, and transitions modifying their quantity. So, the system dynamics is represented by the token game, avoiding the modeler to consider the complete state space of the system. This is useful in particular when the system behavior is characterized by the occurrence of concurrent events.

SAN inherits the features of *Stochastic Petri Nets* (SPN) [10] in particular, where the time to fire of a transition can be a random variable. The *Möbius* tool manages several kinds of probability distributions to be associated with the transitions firing times, and deterministic firing times are available as well. This is a reason why SAN is suitable to model the scenarios under exam in this report, where both stochastic and deterministic events occur. Another advantage of SAN is the presence of a particular modeling primitive called *gate* which allows to express in C code the condition enabling the firing of a transition, or the effect of the firing on the places. In this way, it is possible to set complex firing conditions or effects that would be very complicated (or impossible) to express in a Petri Net only by means of arcs. This allows to simplify the graph structure of the model when we represent complex systems.

Moreover, *Möbius* allows to build models by replicating and joining submodels, by means of a graphical composition model. In this way, the modeler can concentrate its attention on each particular aspect of the system behavior and represent it in form of SAN; then, the SAN models can be easily composed in order to obtain the model of the whole system. Actually composition mechanisms are available also for Generalized

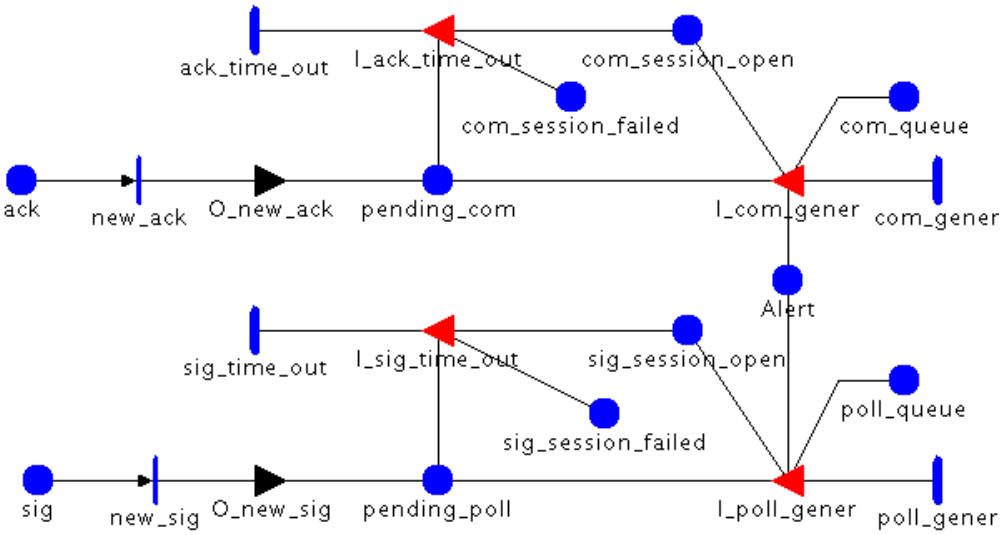

Figure 3: The SAN model *"Control_Centre_Functions"*.

SPN [10], but they require the manual definition of scripts containing the composition operations. This is less intuitive and rapid if compared with the graphical compositional framework in *Möbius*.

5 The SAN models of the scenarios

In our modeling approach, we first model in form of SAN each aspect of the domain in isolation. Then, the SAN models are replicated and joined to obtain the model of the whole domain. Actually several places are shared by the SAN models and they act as points of connection when the models are composed. The model of a scenario is obtained by representing the threat characterizing the scenario in form of SAN, and joining it with the model of the domain, still by superposition over the common places. For the sake of brevity, in this section we briefly describe the SAN models of the domain aspects and of the threats, while all their details can be found in Appendix A.

In the domain under study, the functions of the control centre are the generation of commands and the collection of acknowledgments in the command sessions, and the generation of polls and the collection of signals in the signal sessions (Sec. 2.1). Such functions are represented by the SAN model appearing in Fig. 3 where the upper part concerns the control centre functions during the command sessions while the lower part represents the functions in the signal sessions. The functions performed by a substation are modeled in the SAN model in Fig. 4: the upper part of the model is about the execution of commands, while the lower part of the model concerns the generation of signals.

The transmission of packets can be performed by the communication network $NET1$ or by $NET2$; packets

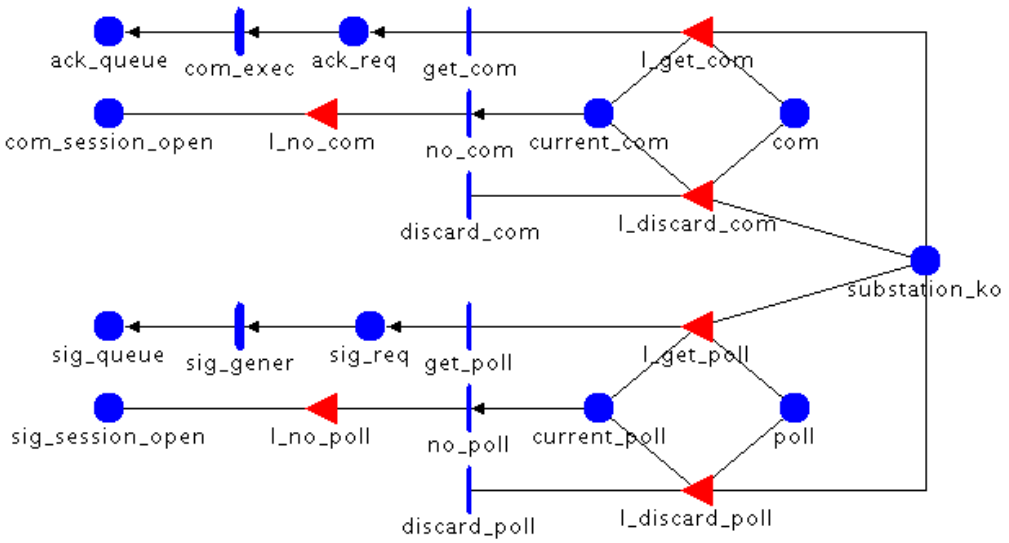

Figure 4: The SAN model *"Substation_Functions"*.

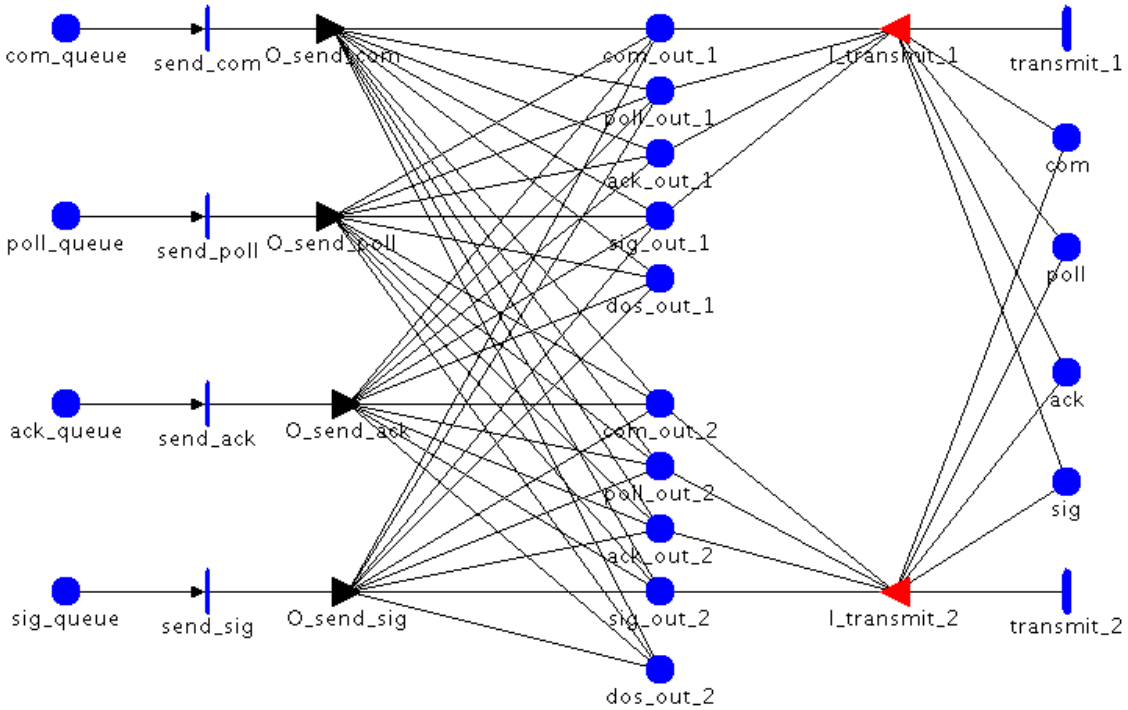

Figure 5: The SAN model *"Packets_Transmission"*.

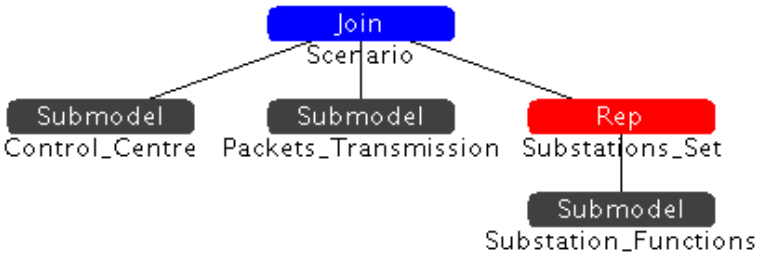

Figure 6: The composed model of the domain.

can be command copies, acknowledgments, poll requests or signals. The SAN model in Fig. 5 represents this situation. The markings of several places in this model represent packets waiting to be transmitted on the available communication network: the tokens inside the places *com_queue* and *poll_queue* represent command copies and poll requests respectively, and they appear also in the SAN model of the control centre (Fig. 3); the tokens inside the places *ack_queue* and *sig_queue* represent acknowledgments and signals respectively, and they appear also in the SAN model of the substation functions in Fig. 4. Other places in the SAN model in Fig. 5 represent instead packets that have been delivered: the markings of the places *ack* and *sig* represent the acknowledgments and the signals respectively, delivered to the control centre; such places appear in the SAN model of the control centre (Fig. 3) as well. The tokens inside the places *com* and *poll* represent the command copies and the poll requests respectively, delivered to the substations; therefore these places belong also to the SAN model of the substation functions (Fig. 4).

Besides representing the packets transmission, the SAN model in Fig. 5 acts as a "bridge" to join the previous SAN models in order to build the model of the whole domain. This is done in Fig. 6 where the SAN model of the substation is replicated 10 times by means of the *Rep* operator (Sec. 4), in order to represent the presence of 10 substations in the domain (Sec. 2.1). The result of the replication and the SAN model of the control centre (Fig. 3) are joined with the SAN model of the packets transmission (Fig. 5), by superposing the common places mentioned above. This is done by means of the *Join* operator (Sec. 4) and generates the model of the domain.

The Scenario 1 is characterized by the occurrence of DoS attacks (Sec. 3) gradually reducing the available bandwidth of the communication network $NET1$ or $NET2$ (Sec. 3.1). The DoS attack is modeled by the SAN in Fig. 7; it contains the place *dos_out* modeling the occupancy of the bandwidth by the packets transmitted by the DoS attack. Since this may affect $NET1$ or $NET2$, two instances of the DoS attack model are composed with the model of the domain in order to obtain the model of the Scenario 1 (Fig. 8). One instance represent the DoS attack to $NET1$, so its place *dos_out* corresponds to the place *dos_out_1* in the SAN model of the packets transmission (Fig. 5). The other instance concerns the attack to $NET2$; therefore its place

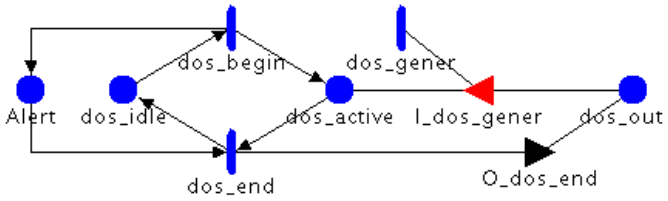

Figure 7: The SAN model *"DoS_attack"*.

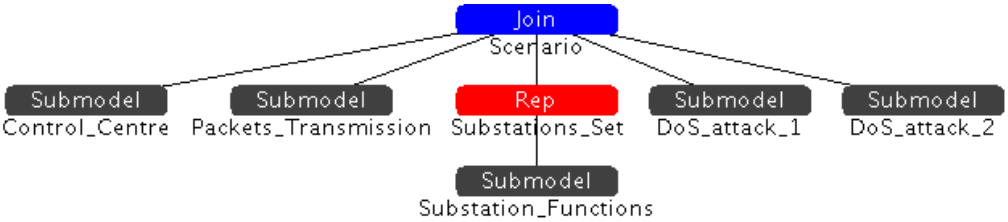

Figure 8: The composed model of the Scenario 1.

dos_out corresponds to the place *dos_out_2* of the packets transmission model. In this way, the model in Fig. 5 takes into account the bandwidth consumption also by means of the DoS packets, and acts as a bridge also to include the DoS attack in the scenario model.

In the Scenario 2, the communication may be compromised by the unavailability of the substations (Sec. 3), caused by the failure of their internal components (Sec. 3.2). The SAN model in Fig. 9 represents the failure and the repair of the substation components; such model consists of the conversion into SAN, of the FT model in Fig. 2, with the addition of the repair actions, each involving a single component of the substation. In particular, this SAN model contains the place *substation_ko* indicating if the substation is currently unavailable or not. The composed model of the Scenario 2 in Fig. 10 is derived from the domain model (Fig. 2) in this way: before the replication, the SAN model of the substation functions (Fig. 4) is joined with the SAN model of the substation failure and repair (Fig. 9), by superposition over the common place *substation_ko*. In this way, in the resulting model of the substation, its functions are disabled if such place is marked (the substation is unavailable). Then, such model is replicated in order to represent the set of 10 substations in the domain.

Finally, the Scenario 3 takes into account both the DoS attacks and the substations failures. So, the its composed model (Fig. 11) is obtained from the model of the domain by including two instances of the DoS attack SAN model, and the SAN model of the substation failure and repair. Such models are joined with those of the domain aspects in the same ways as in the case of the Scenarios 1 and 2 respectively.

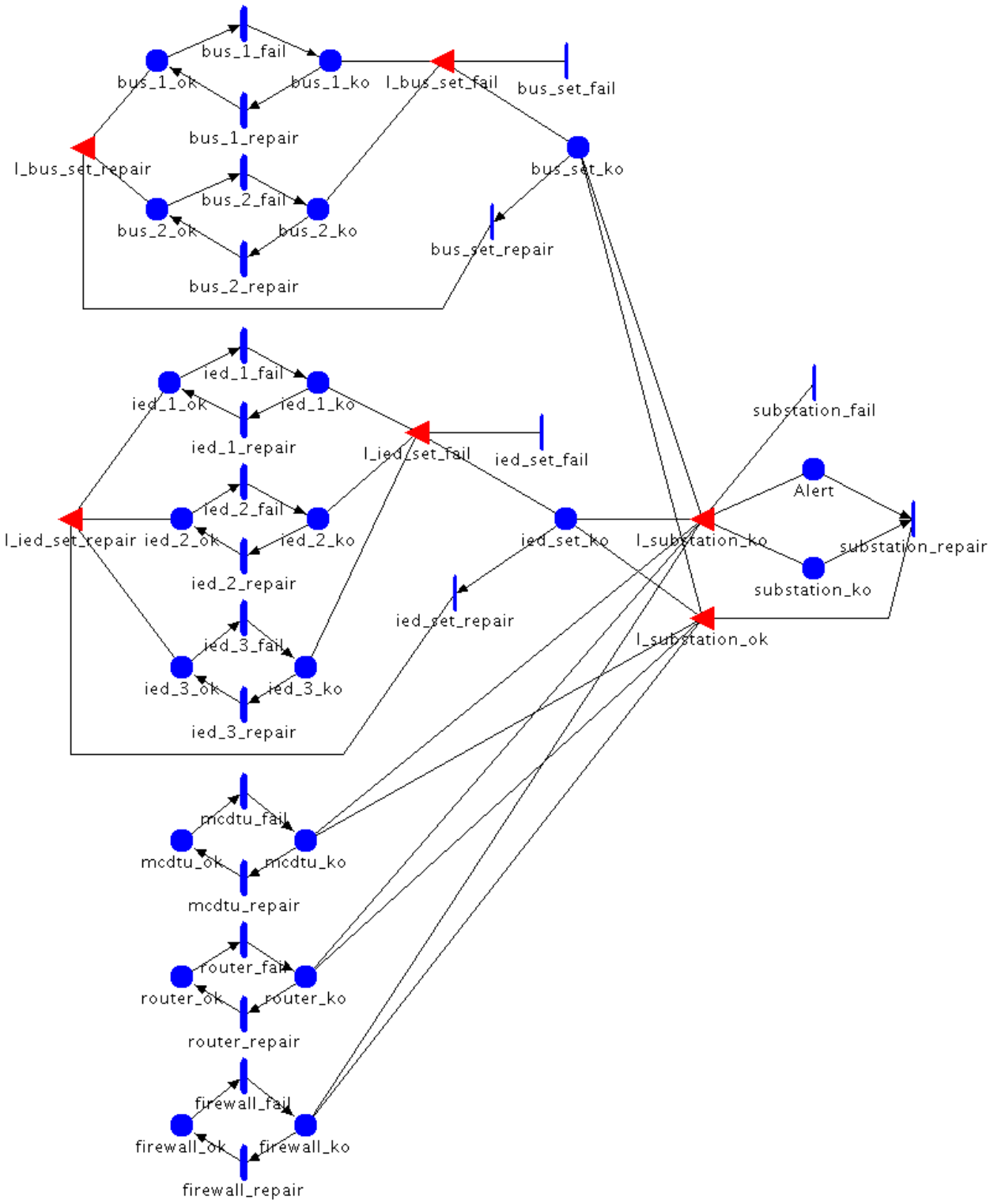

Figure 9: The SAN model "Substation_Failure_and_Repair".

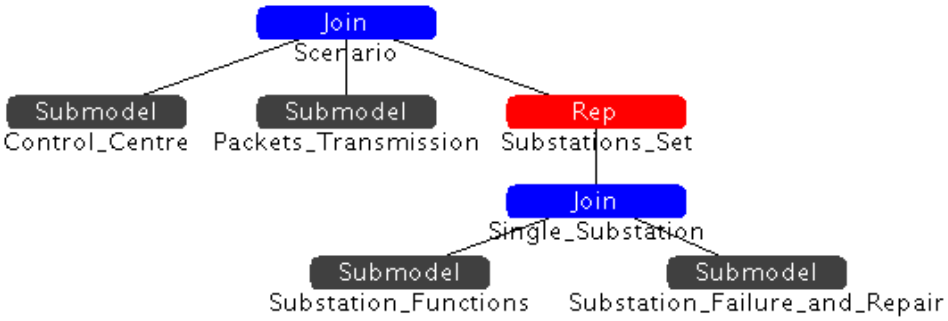

Figure 10: The composed model of the Scenario 2.

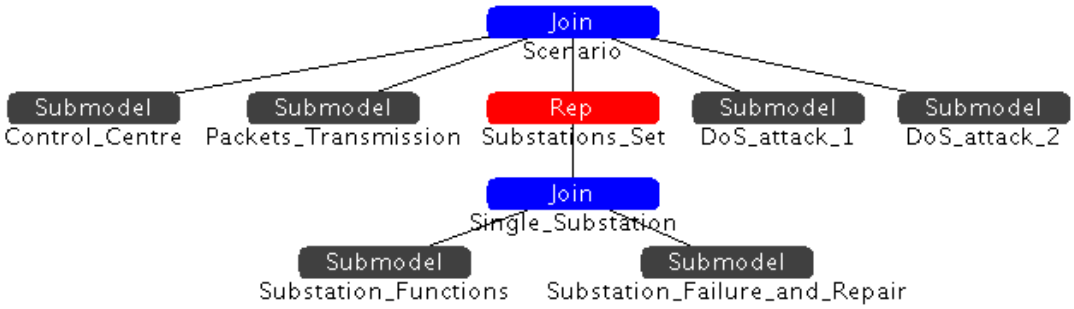

Figure 11: The composed model of the Scenario 3.

6 The scenarios simulation

For each scenario model described in the previous section, 10000 simulation batches have been performed by means of *Möbius*, setting a confidence level of 0.95, and a relative confidence interval of 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 signal sessions at a certain time.

The functions expressing such measures in terms of place markings are reported in Appendix B. All measures are computed for a mission time varying between 0 and 10000 h. The values of $Pr_{com}(t)$ returned by the simulation in each scenario are reported in Tab. 4 and are depicted in Fig. 12.a. Tab. 4 and Fig. 12.b show the results obtained for $Pr_{sig}(t)$ in each scenario.

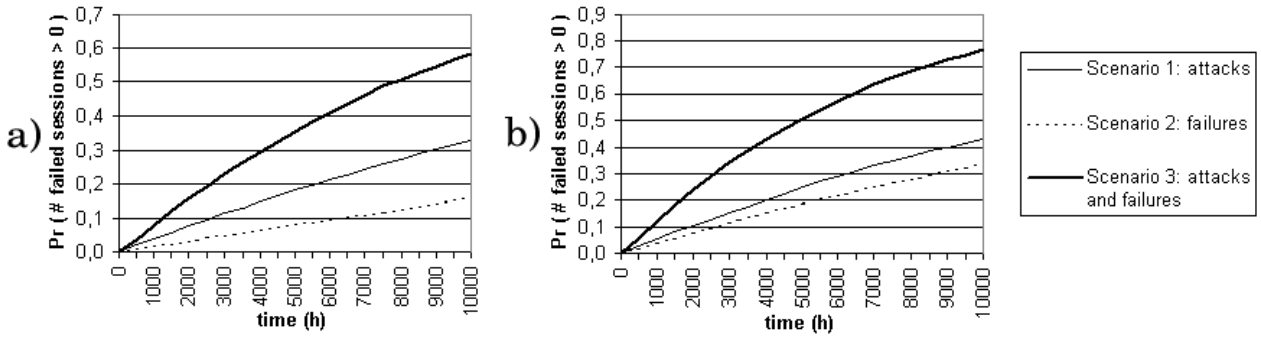

Figure 12: a) $Pr_{com}(t)$ (Tab. 4). b) $Pr_{sig}(t)$ (Tab. 4).

	$Pr_{com}(t)$			$Pr_{sig}(t)$		
time	Scenario 1	Scenario 2	Scenario 3	Scenario 1	Scenario 2	Scenario 3
1000 h	3,850E-02	1,530E-02	7,770E-02	5,310E-02	3,750E-02	1,201E-01
2000 h	7,720E-02	3,230E-02	1,571E-01	1,037E-01	7,750E-02	2,397E-01
3000 h	1,159E-01	4,890E-02	2,280E-01	1,524E-01	1,152E-01	3,420E-01
4000 h	1,469E-01	6,400E-02	2,944E-01	1,994E-01	1,529E-01	4,292E-01
5000 h	1,799E-01	8,000E-02	3,559E-01	2,474E-01	1,854E-01	5,070E-01
6000 h	2,111E-01	9,390E-02	4,089E-01	2,898E-01	2,178E-01	5,732E-01
7000 h	2,418E-01	1,101E-01	4,608E-01	3,287E-01	2,499E-01	6,326E-01
8000 h	2,727E-01	1,260E-01	5,050E-01	3,670E-01	2,783E-01	6,829E-01
9000 h	3,010E-01	1,430E-01	5,451E-01	4,003E-01	3,072E-01	7,273E-01
10000 h	3,283E-01	1,596E-01	5,853E-01	4,327E-01	3,371E-01	7,655E-01

Table 4: $Pr_{com}(t)$ (Fig. 12.a) and $Pr_{sig}(t)$ (Fig. 12.b).

Both Fig. 12.a and Fig. 12.b show that according to the event occurrence times specified in Sec. 2 and the SAN models described in Sec. 5, the DoS attacks (Scenario 1) determine an higher probability of command or signal session failure, with respect to the substation failures (Scenario 2). In the Scenario 1, if only one communication network is under attack, its bandwidth is gradually reduced, while the bandwidth of the other network is completely available for normal communication. In this situation, the global bandwidth of both networks is enough to transmit the packets concerning a single session. Some packets may be lost instead, if a command session is running in parallel with a signal session; in this case, both networks are necessary to transmit all the packets, as described in Sec. 2.2: the residual available bandwidth of the network under attack may not be enough to transmit all the packets that exceeds the bandwidth of the other network. This will determine the failure of command sessions because command copies or acknowledgments are not delivered, or the failure of signal sessions because poll requests or signals are not delivered.

Still in the Scenario 1, if both communication networks are under attack at the same time, a session may fail also if it is not running in parallel with another one. This happens because the residual available bandwidths of both networks may not be enough to transmit all the packets. In the Scenario 2 instead, a command or signal session fails if at least two substations are unavailable at the same time due to internal

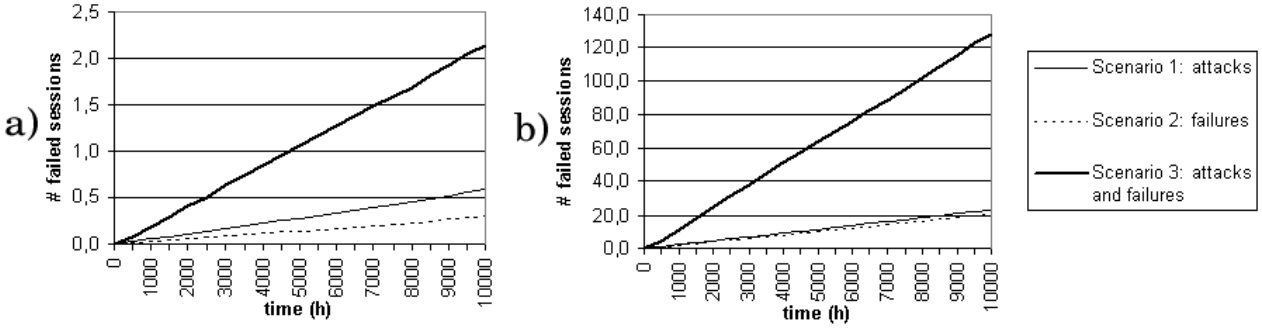

Figure 13: a) $Num_{com}(t)$ (Tab. 5). b) $Num_{sig}(t)$ (Tab. 5). During 10000 h the mean number of command sessions is about 1665, the number of signal session is about 112500.

	$Num_{com}(t)$			$Num_{sig}(t)$		
time	Scenario 1	Scenario 2	Scenario 3	Scenario 1	Scenario 2	Scenario 3
1000 h	5,500E-02	3,130E-02	1,8390E-01	2,172E+00	1,942E+00	1,080E+01
2000 h	1,130E-01	5,980E-02	4,1050E-01	4,453E+00	3,931E+00	2,468E+01
3000 h	1,731E-01	9,000E-02	6,2860E-01	6,800E+00	6,049E+00	3,793E+01
4000 h	2,280E-01	1,163E-01	8,4240E-01	9,004E+00	8,044E+00	5,113E+01
5000 h	2,836E-01	1,452E-01	1,0619E+00	1,140E+01	1,033E+01	6,379E+01
6000 h	3,406E-01	1,705E-01	1,2724E+00	1,368E+01	1,216E+01	7,610E+01
7000 h	3,980E-01	2,015E-01	1,4863E+00	1,607E+01	1,434E+01	8,911E+01
8000 h	4,643E-01	2,341E-01	1,6925E+00	1,872E+01	1,641E+01	1,020E+02
9000 h	5,209E-01	2,712E-01	1,9121E+00	2,112E+01	1,876E+01	1,150E+02
10000 h	5,767E-01	3,057E-01	2,1284E+00	2,339E+01	2,116E+01	1,281E+02

Table 5: $Num_{com}(t)$ (Fig. 13.a) and $Num_{sig}(t)$ (Fig. 13.b).

failures. Since a substation in failure condition does not reply to commands and polls, in this situation, at least 2 acknowledgments or 2 signals will be missing when the command (signal) session is closed, determining the session failure (Sec. 3).

The fact that DoS attacks affect the communication reliability more than the substations unavailabilities is confirmed in terms of number of failed sessions, by the results obtained for the measures $Num_{com}(t)$ (Tab. 5 and Fig. 13.a) and $Num_{sig}(t)$ (Tab. 5 and Fig. 13.b).

In the Scenario 3, a command or a signal session can fail due to a DoS attack, to the substations failure, or to both causes. For instance, a command session may fail because one acknowledgment is missing because a failed substation has not executed the command, and another acknowledgment is missing because the communication networks are under DoS attack and the acknowledgment becomes lost. Actually, observing Fig. 12.a and Fig. 12.b, we can notice that the values of both $Pr_{com}(t)$ and $Pr_{sig}(t)$ in the Scenario 3 are about the sum of the same probabilities in the Scenario 1 and in the Scenario 2.

7 Conclusions

The report examined the communication between the control centre and the substations of a distribution grid of the EPS. Scenarios characterized by the occurrences of DoS attacks and substations failures, have been evaluated: we have obtained that the first type of threat has a higher negative influence on the communication reliability. This evaluation has been performed by modeling and simulating the domain and the scenarios in form of SAN. The use of this formalism allowed to model both the stochastic and the deterministic events realizing the communication both in normal conditions and in presence of threats. Besides this, the SAN formalism and the *Möbius* tool in particular, allowed to build the models of the scenarios, by composition of several submodels representing particular aspects of the domain or of the threats. Scenarios in the same domain, but characterized by other threats (intrusions in the communication and unavailability of the communication network), are evaluated in [1, 2].

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 (accepted for publication).*

[2] D. Codetta-Raiteri and R. Nai. SAN models of communication scenarios inside the Electrical Power System. Technical Report TR-INF-2009-07-05-UNIPMN, Dip. di Informatica, Univ. del Piemonte Orientale, July 2009. http://www.di.unipmn.it.

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

[4] D. Cerotti, D. Codetta-Raiteri, S. Donatelli, C. Brasca, G. Dondossola, and F. Garrone. UML diagrams supporting domain specification inside the CRUTIAL project. *Lecture Notes in Computer Science*, 5141:106–123, November 2008.

[5] 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, January 2007.

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

[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] 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.

[9] W. G. Schneeweiss. *The Fault Tree Method.* LiLoLe Verlag, 1999.

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

A The SAN models in details

This appendix provides all the details about the SAN models in Sec. 5, each representing a particular aspect of the domain or the threats involved in the scenarios.

A.1 Modeling 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 networks $NET1$ and $NET2$, and the substation functions (execution of commands, generation of acknowledgments, generation of signals).

A.1.1 Modeling 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 firing (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. A.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 firing 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 firing 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 modeled 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 modeled 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* fires 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 firing 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 fires, 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*.

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

Tab. 6 shows the firing times and the input or output gates associated with each activity in the SAN model *"Control_Centre_Functions"* in Fig. 3.

A.1.2 Modeling the substation functions

The functions performed by a substation are modeled 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 modeled 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 lower 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* 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 modeled 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 functions of the substation (execution of commands and generation of signals) can not be performed if the substation is currently failed. The failed state of the substation is modeled by the presence of one token inside the place *substation_ko* which is present also in the SAN model called *"Substation_Failure_and_Repair"* in Fig. 9 considering the failure and repair of the substation. If the place *substation_ko* becomes marked (see Sec. A.3), then both the immediate activities *get_com* and *get_poll* are disabled, while both the immediate activities *discard_com* and *discard_poll* are enabled according to the predicate defined in the input gates *I_discard_com* and *I_discard_poll* respectively. In this situation, one token in the place *com* or in the place *poll* is consumed if they are marked, but no acknowledgments or signals are generated. In this way, we model that in case of substation failure, though a command copy or a poll request is received by the substation,

Activity:	*com_gener*
type:	stochastic
mean time to fire:	6 h
firing 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;
	pending_com->Mark()=10;
	com_queue->Mark()=10;

Activity:	*new_ack*
type:	immediate
output gate:	*O_new_ack*
output gate function:	if (pending_com->Mark() > 0) pending_com->Mark()–;
Activity:	*ack_time_out*
type:	deterministic
time to fire:	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;
	if (pending_com->Mark() > 1)
	com_session_failed->Mark()++;

Activity:	*poll_gener*
type:	deterministic
time to fire:	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;
	pending_poll->Mark()=10;
	poll_queue->Mark()=10;

Activity:	*new_sig*
type:	immediate
output gate:	*O_new_sig*
output gate function:	if (pending_poll->Mark() > 0) pending_poll->Mark()–;
Activity:	*sig_time_out*
type:	deterministic
input gate:	*I_ack_time_out*
input gate predicate:	sig_session_open->Mark()==1

output gate function:	sig_session_open->Mark()=0;
	if (pending_poll->Mark() > 1)
	sig_session_failed->Mark()++;

Table 6: The activities in the SAN model *"Control_Centre_Functions"* (Fig. 3).

Activity:	*get_com*
type:	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*
type:	stochastic
mean time to fire:	2.77778E-04 h
firing rate:	3600 h^{-1}
Activity:	*discard_com*
type:	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*
type:	immediate
input gate:	*I_no_com*
input gate predicate:	com_session_open->Mark()==0
Activity:	*get_poll*
type:	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*
type:	stochastic
mean time to fire:	2.77778E-04 h
firing rate:	3600 h^{-1}
Activity:	*discard_poll*
type:	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*
type:	immediate
input gate:	*I_no_poll*
input gate predicate:	sig_session_open->Mark()==0

Table 7: The activities in the SAN model *"Substation_Functions"* (Fig. 4).

there is no reply by the substation.

Tab. 7 summarizes the activities inside the model *"Substation_Functions"* in Fig. 4, including the predicates and the functions of the gates ruling the firing of the activities.

A.1.3 Modeling 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 or signals. The SAN model *"Packets_Transmission"* in Fig. 5 represents this situation. The markings of the several places in this model represent packets waiting to be transmitted on the available communication network: the place *com_queue* and the place *poll_queue* concern command copies and poll requests respectively, and they appear also in the SAN model

"Control_Centre_Functions" in Fig. 3. The place *ack_queue* and the place *sig_queue* concern acknowledgments and signals respectively, and they appear also in the SAN model *"Substation_Functions"* in Fig. 4.

We suppose that the bandwidth of each communication network is equal to 16 $kbit/sec$ and that the transmission of each packet requires to consume 1 $kbit/sec$ of the bandwidth (Sec. 2.2). 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 *dos_out_1* represent the number of command copies, poll requests, acknowledgments, signals and DoS packets (see Sec. A.2) respectively that are currently under transmission by $NET1$. When a token appears in *com_queue*, the immediate activity *send_com* fires 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 *dos_out_1* is less than 16 ($16kbit/sec$ is the bandwidth of $NET1$). If so, enough bandwidth is available to transmit the command copies, 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, then no bandwidth is currently available on $NET1$ (this may happen in case of DoS attack (see Sec. A.2)): 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 *dos_out_2* is less than 16, 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 presence of acknowledgments, polls and signals to be transmitted, is modeled by the marking of the places *poll_queue*, *ack_queue*, *sig_queue* respectively. The direction of suck kinds of packets toward $NET1$ or $NET2$ is modeled in a way similar to the command copies direction: the output gates *O_send_poll*, *O_send_ack*, *O_send_sig* perform the same checks and have the same effect of *O_send_com*, in case of firing of the activities *send_poll*, *send_ack*, *send_sig* respectively, due to the presence of a token inside the places *poll_queue*, *ack_queue*, *sig_queue* respectively.

The transmission of the packets by $NET1$ is modeled by the stochastic activity *transmit_1* whose firing 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

Activity:	*send_com*
type:	immediate
output gate:	*O_send_com*

output gate predicate:	if (com_out_1->Mark() + poll_out_1->Mark() + ack_out_1->Mark() + sig_out_1->Mark() + dos_out_1->Mark() < 16)
	  com_out_1->Mark()++;
	else
	  if (com_out_2->Mark() + poll_out_2->Mark() + ack_out_2->Mark() + sig_out_2->Mark() + dos_out_2->Mark() < 16)
	  com_out_2->Mark()++;

Activity:	*send_poll*
type:	immediate
output gate:	*O_send_poll*

output gate predicate:	if (com_out_1->Mark() + poll_out_1->Mark() + ack_out_1->Mark() + sig_out_1->Mark() + dos_out_1->Mark() < 16)
	  poll_out_1->Mark()++;
	else
	  if (com_out_2->Mark() + poll_out_2->Mark() + ack_out_2->Mark() + sig_out_2->Mark() + dos_out_2->Mark() < 16)
	  poll_out_2->Mark()++;

Activity:	*send_ack*
type:	immediate
output gate:	*O_send_ack*

output gate predicate:	if (com_out_1->Mark() + poll_out_1->Mark() + ack_out_1->Mark() + sig_out_1->Mark() + dos_out_1->Mark() < 16)
	  ack_out_1->Mark()++;
	else
	  if (com_out_2->Mark() + poll_out_2->Mark() + ack_out_2->Mark() + sig_out_2->Mark() + dos_out_2->Mark() < 16)
	  ack_out_2->Mark()++;

Activity:	*send_sig*
type:	immediate
output gate:	*O_send_sig*

output gate predicate:	if (com_out_1->Mark() + poll_out_1->Mark() + ack_out_1->Mark() + sig_out_1->Mark() + dos_out_1->Mark() < 16)
	  sig_out_1->Mark()++;
	else
	  if (com_out_2->Mark() + poll_out_2->Mark() + ack_out_2->Mark() + sig_out_2->Mark() + dos_out_2->Mark() < 16)
	  sig_out_2->Mark()++;

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

control centre; *sig* is the same place present in the SAN model *"Control_Centre_Functions"* (Fig. 3).

The transmission of packets by $NET2$ is modeled 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).

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

Activity:	*transmit_1*
type:	stochastic
mean time to fire:	2.77778E-04 h
firing 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)

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; }

Activity:	*transmit_2*
type:	stochastic
mean time to fire:	2.77778E-04 h
firing 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)

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; }

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

Activity:	*dos_begin*
type:	stochastic
mean time to fire:	720 h
firing rate:	1.38889E-03 h^{-1}
Activity:	*dos_end*
type:	stochastic
mean time to fire:	12 h
firing rate:	8.33333E-02 h^{-1}
output gate:	*O_dos_end*
output gate function:	dos_out->Mark()=0;
Activity:	*dos_gener*
type:	stochastic
mean time to fire:	0.1875 h
firing rate:	5.33333 h^{-1}
input gate:	*I_dos_gener*
input gate predicate:	(dos_active->Mark() > 0) && (dos_out->Mark() < 16)
input gate function:	dos_out->Mark()++;

Table 10: The activities in the SAN model *"DoS_attack"* (Fig. 7).

A.2 Modeling the DoS attack

The DoS attack gradually reducing the available bandwidth of the communication network $NET1$ or $NET2$, is modeled by the SAN called *"DoS_attack"* in Fig. 7. In this model, when no attack is running, the place *dos_idle* contains one token. In this situation, the stochastic activity *dos_begin* can fire moving the token from *dos_idle* into the place *dos_active*. In this way, we model that an attack has begun. While the place *dos_active* is marked, the stochastic activity *dos_gener* can fire several times, increasing the marking of the place *dos_out* by one, each time. Such place represents the bandwidth occupancy by the DoS packets, and corresponds in the SAN model *"Packets_Transmission"* (Fig. 5) to the place *dos_out_1* (in case of DoS attack to $NET1$) or to the place *dos_out_2* (in case of DoS attack to $NET2$). The marking of the place *dos_out* can not exceed 16 tokens, corresponding to the maximum bandwidth occupancy (input gate *I_dos_gener*). The presence of tokens inside *dos_out* may cause the direction of packets toward the communication network $NET1$ or $NET2$ in the model *"Packets_Transmission"* (Fig. 5) (see Sec. A.1.3).

The end of the attack is modeled by the stochastic activity *dos_end* whose firing has a double effect: moving the token inside the place *dos_active* into the place *dos_idle*, and removing any token inside the place *dos_out* (by the output gate *O_dos_end*). In this way, the bandwidth of $NET1$ or $NET2$, occupied during the attack, becomes available again.

Tab. 10 reports the firing times, the gate predicates and functions concerning the activities in the SAN model *"DoS_attack"* in Fig. 7.

A.3 Modeling the substation failure and repair

In this section, we describe the SAN model *"Substation_Failure_and_Repair"* (Fig. 9) representing the failure and the repair of the substation components; such model consists of the conversion into SAN form of the FT

model in Fig. 2, with the addition of the repair actions, each involving a single component of the substation.

Let us consider first the component $IED1$ (see Sec. 3.2): the place *ied_1_ok* is initially marked with one token in order to model that the component $IED1$ is working. Such token can be moved into the place *ied_1_ko* by the stochastic activity *ied_1_fail*; in this way, we model the current failed state of the component $IED1$. The repair of this component is modeled by the stochastic activity *ied_1_repair* moving the token from the place *ied_1_ko* into the place *ied_1_ok*. The failure and the repair of all the other components of the substation is modeled in a similar way.

The immediate activity *ied_set_fail* fires when the places *ied_1_ko*, *ied_2_ko*, *ied_3_ko* are all marked; the effect of the firing is the presence of one token inside the place *ied_set_ko* and this means that all the IED components are failed. The condition enabling *ied_set_fail* to fire, and the effect of its firing are defined in the input gate *I_ied_set_fail*. If at least one IED component is repaired, the immediate activity *ied_set_repair* fires removing the token inside the place *ied_set_ko*, as specified in the input gate *I_ied_set_repair*.

In a similar way, the place *bus_set_ko* becomes marked as a consequence of the firing of the immediate activity *bus_set_fail* when both the place *bus_1_ko* and *bus_2_ko* are marked, according to the input gate *I_bus_set_ko*. The token inside *bus_set_ko* indicates that both buses are failed, and is removed by the immediate activity *bus_set_repair* when at least one bus is repaired, according to the input gate *I_bus_set_repair*.

The failure of the substation is modeled by the presence of one token inside the place *substation_ko*. This place becomes marked if the activity *substation_fail* fires; this happens at least one of the following conditions holds: the place *bus_set_ko* is marked (all the buses are currently failed); the place *ied_set_ko* is marked (all the IEDs are currently failed); the place *mcdtu_ko* is marked (the MCDTU is currently failed); the place *router_ko* is marked (the router is currently failed); the place *firewall_ko* is marked (the firewall is currently failed). This is specified in the input gate *I_substation_fail*. The substation turns available again when the immediate activity *substation_repair* fires and consequently the place *substation_ko* becomes empty. Such firing can occur when all the following conditions hold: the place *bus_set_ko* is empty (at least one bus is currently working); the place *ied_set_ko* is empty (at least one IED is currently working); the place *mcdtu_ko* is empty (the MCDTU is currently working); the place *router_ko* is empty (the router is currently working); the place *firewall_ko* is empty (the firewall is currently working). This is specified in the input gate *I_substation_repair*.

The activities in the SAN model *"Substation_Failure_and_Repair"* in Fig. 9 are detailed in Tab. 11 and in Tab. 12.

Activity:	*bus_1_fail*	Activity:	*bus_1_repair*
type:	stochastic	type:	stochastic
mean time to fire:	4380 h	mean time to fire:	24 h
firing rate:	2.28311E-04 h^{-1}	firing rate:	4.16667E-02 h^{-1}
Activity:	*bus_2_fail*	Activity:	*bus_2_repair*
type:	stochastic	type:	stochastic
mean time to fire:	4380 h	mean time to fire:	24 h
firing rate:	2.28311E-04 h^{-1}	firing rate:	4.16667E-02 h^{-1}
Activity:	*ied_1_fail*	Activity:	*bus_1_repair*
type:	stochastic	type:	stochastic
mean time to fire:	4380 h	mean time to fire:	48 h
firing rate:	2.28311E-04 h^{-1}	firing rate:	2.08333E-02 h^{-1}
Activity:	*ied_2_fail*	Activity:	*bus_2_repair*
type:	stochastic	type:	stochastic
mean time to fire:	4380 h	mean time to fire:	48 h
firing rate:	2.28311E-04 h^{-1}	firing rate:	2.08333E-02 h^{-1}
Activity:	*ied_3_fail*	Activity:	*bus_3_repair*
type:	stochastic	type:	stochastic
mean time to fire:	4380 h	mean time to fire:	48 h
firing rate:	2.28311E-04 h^{-1}	firing rate:	2.08333E-02 h^{-1}
Activity:	*mcdtu_fail*	Activity:	*mcdtu_repair*
type:	stochastic	type:	stochastic
mean time to fire:	8760 h	mean time to fire:	12 h
firing rate:	1.14155E-04 h^{-1}	firing rate:	8.33333E-02 h^{-1}
Activity:	*router_fail*	Activity:	*router_repair*
type:	stochastic	type:	stochastic
mean time to fire:	17520 h	mean time to fire:	6 h
firing rate:	5.70776E-05 h^{-1}	firing rate:	1.66667E-01 h^{-1}
Activity:	*firewall_fail*	Activity:	*firewall_repair*
type:	stochastic	type:	stochastic
mean time to fire:	17520 h	mean time to fire:	6 h
firing rate:	5.70776E-05 h^{-1}	firing rate:	1.66667E-01 h^{-1}

Table 11: The activities in the SAN model *"Substation_Failure_and_Repair"* (Fig. 9).

Activity:	*bus_set_fail*
type:	immediate
input gate:	*I_bus_set_fail*
input gate predicate:	(bus_1_ko->Mark()==1) && (bus_2_ko->Mark()==1) && (bus_set_ko->Mark()==0)
input gate function:	bus_set_ko->Mark()=1;
Activity:	*bus_set_repair*
type:	immediate
input gate:	*I_bus_set_repair*
input gate predicate:	(bus_1_ok->Mark()==1) || (bus_2_ok->Mark()==1)
Activity:	*ied_set_fail*
type:	immediate
input gate:	*I_ied_set_fail*
input gate predicate:	(ied_1_ko->Mark()==1) && (ied_2_ko->Mark()==1) && (ied_1_ko->Mark()==1) && (bus_set_ko->Mark()==0)
input gate function:	ied_set_ko->Mark()=1;
Activity:	*ied_set_repair*
type:	immediate
input gate:	*I_ied_set_repair*
input gate predicate:	(ied_1_ok->Mark()==1) || (ied_2_ok->Mark()==1) || (ied_3_ok->Mark()==1)
Activity:	*substation_fail*
type:	immediate
input gate:	*I_substation_ko*
input gate predicate:	(substation_ko->Mark() == 0) && (bus_set_ko->Mark()==1 || ied_set_ko->Mark()==1 || mcdtu_ko->Mark()==1 || router_ko->Mark()==1 || firewall_ko->Mark()==1)
input gate function:	substation_ko->Mark() = 1; Alert->Mark()++;
Activity:	*substation_repair*
type:	immediate
input gate:	*I_substation_ok*
input gate function:	(bus_set_ko->Mark()==0) && (ied_set_ko->Mark()==0) && (mcdtu_ko->Mark()==0) && (router_ko->Mark()==0) && (firewall_ko->Mark()==0)

Table 12: The activities in the SAN model *"Substation_Failure_and_Repair"* (Fig. 9).

B Measures and functions

In Sec. 6, we have reported the simulation results obtained for the following measures:

1. $Pr_{com}(t)$: the probability that at least one command session has failed at a certain time (Tab. 4 and Fig. 12.a);
2. $Pr_{sig}(t)$: the probability that at least one signal session has failed at a certain time (Tab. 4 and Fig. 12.b);
3. $Num_{com}(t)$: mean number of failed command sessions at a certain time (Tab. 5 and Fig. 13.a);
4. $Num_{sig}(t)$: mean number of failed signal sessions at a certain time (Tab. 5 and Fig. 13.b).

In this appendix, we provide the functions expressing such measures in terms of place markings:

1. 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_Activity->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.

2. 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_Activity->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 $Pr_{sig}(t)$.

3. 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_Activity->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 ($Num_{com}(t)$).

4. 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_Activity->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.