Analysis of a Redundant Architecture for Critical Infrastructure Protection

Alessandro Daidone[1], Silvano Chiaradonna[2], Andrea Bondavalli[1], and Paulo Veríssimo[3]

[1] University of Florence, viale Morgagni 65, I-50134, Italy
daidone@dsi.unifi.it, bondavalli@unifi.it
[2] ISTI-CNR, via Moruzzi 1, I-56124, Italy
silvano.chiaradonna@isti.cnr.it
[3] University of Lisbon, Campo Grande 1749-016, Lisbon, Portugal
pjv@di.fc.ul.pt

Abstract. Critical infrastructures like the power grid are emerging as collection of existing separated systems of different nature which are interconnected together. Their criticality becomes more and more evident as the damage and the risks deriving from wrong behaviors (both accidental and intentionally caused) are increasing. It is becoming evident that existing (legacy) subsystem must be interconnected together following some disciplined and controlled way. This is one of the challenges taken by the European Project CRUTIAL, where an infrastructure architecture seen as a WAN of LANs is being proposed, where LANs confine existing sub-systems, protected by special interconnection and filtering devices (CIS - CRUTIAL Information Switches). Previous work led to the definition of the CIS internal and interconnection architecture, so that a set of CIS can collectively ensure that the computers controlling the physical process correctly exchange information despite accidents and malicious attacks. CIS resilience is achieved thanks to replication for intrusion tolerance and replica recovery for self-healing.
This chapter analyzes the redundant architecture of the CIS, with a set of objectives: identifying the relevant parameters of the architecture; evaluating how effective is the trade-off between proactive and reactive recoveries; and finding the best parameter setup. Two measures of interest were identified, a model of the recovery strategy was constructed and the quantitative behavior of the recovery strategy was analyzed. The impact of the detection coverage, of the intrusions and of the number of CIS replicas was analyzed and discussed. The directions for refining and improving the recovery strategy were proposed.

1 Introduction

Critical infrastructures (e.g., the power grid) are basically physical processes controlled by computers interconnected by networks [1]. Some years ago those systems were highly isolated and hence secure against most security threats. During the last years the Information and Communications Technology (ICT)

part of those critical infrastructures evolved in several aspects: i) hardware and software devices (station computers, networks, protocols,...) are no longer ad-hoc and proprietary, instead standard components (COTS[4]) are used; ii) most of the station computers are connected to corporate networks and to the Internet. Therefore these infrastructures are nowadays greatly exposed to cyber-attacks coming from the Internet [2, 3], so they have a level of vulnerability similar to other systems connected to the Internet, but the socio-economic impact of their failure can be huge. This scenario, reinforced by several recent incidents, is generating a great concern about the security of these infrastructures, especially at government level [4].

A reference architecture [5] was recently proposed to protect the power grid in the context of the CRUTIAL[5] EU-IST project. Since the power grid is formed by facilities (power transformation substations, corporate offices, etc.) interconnected by a wider-area network (WAN), [5] proposes to represent facilities using protected LANs interconnected by a WAN, leading to the WAN-of-LANs architecture. Using such an architecture, the problem of protecting the power grid (and similar critical infrastructures) is reduced to the problem of protecting LANs from the WAN or other LANs.

In the CRUTIAL reference architecture each LAN is connected to the WAN through a special interconnection and filtering device, the CIS (CRUTIAL Information Switch), which ensures that both the incoming and outgoing traffic satisfies the security policy defined to protect the infrastructure (the so called CIS Protection Service). A CIS is hence a kind of improved firewall that works at the application layer and that is intrusion tolerant. CIS resilience is achieved thanks to replication for intrusion tolerance and replica recovery for self-healing [6, 7]. Replication is used in order to guarantee system correct operation when some replicas are compromised. Rejuvenation is instead used to remove the effects of malicious attacks aiming to compromise some replicas and to break the system. The replica rejuvenation strategy, PRRW (Proactive-Reactive Recovery Wormhole), is based both on periodic (proactive) recoveries and on event triggered (reactive) recoveries, seeking perpetual unattended correct operation.

The proactive-reactive recovery strategy aims to both increase CIS dependability and guarantee CIS availability, despite of faults, intrusions and recoveries. In particular, recoveries have beneficial effects (e.g., reactive recoveries rejuvenate replicas detected as incorrect), but also negative effects (e.g., the proactive recovery of a correct replica makes the replica unavailable for the whole duration of the recovery). The key property of the PRRW strategy is that, as long as the fault exhibited by the replica is detectable, this replica will be recovered as soon as possible, ensuring that there is always an amount of replicas available to sustain correct operation [7].

This chapter analyzes the redundant architecture of the CIS, with a set of objectives: evaluating how effective is the trade-off between proactive and reactive recoveries, identifying the relevant parameters of the architecture and finding the

[4] Commercial Off-The-Shelf components
[5] CRitical UTility InfrastructurAL resilience: http://crutial.cesiricerca.it

best parameter setup. Two dependability and availability measures of interest were identified. A model of the recovery strategy was constructed in order to analyze the quantitative behavior of the recovery strategy. The impact of the detection coverage, of the intrusions and of the number of CIS replicas on the measures of interest was analyzed and discussed. The directions for refining and improving the recovery strategy were proposed.

The rest of the chapter is organized as follows. Section 2 gives an overview of the reference architecture used in CRUTIAL; Section 3 gives an overview both of the CIS and the PRRW recovery strategy; Section 4 presents the models and the quantitative analysis of the PRRW strategy; Section 5 identifies the directions for improvements and refinements on the recovery strategy. Finally, concluding remarks are presented in Sect. 6.

2 CRUTIAL Reference Architecture Overview

The infrastructure architecture in CRUTIAL is modeled as a WAN-of-LANs [5]. All the Information and Communications Technology (ICT) parts necessary for the control of the whole power grid[6] are logically grouped in substations and finally in local area networks (LANs). LANs are interconnected by a global interconnection network, called WAN. The WAN is a logical entity owned and operated by the critical information infrastructure operator companies, which may or not use parts of public network as physical support. All traffic originates from and goes to a LAN, so packets are switched by the WAN through substation gateways called CRUTIAL Information Switches (CIS).

CIS collectively act as a set of servers providing distributed services aimed to control both the command and information flow among the ICT parts of the critical infrastructure, securing a set of necessary system-level properties. This set of servers must be intrusion-tolerant, prevent resource exhaustion providing perpetual operation, and be resilient against assumption coverage uncertainty, providing graceful degradation or survivability. An assumed number of CIS can be corrupted; in consequence, a logical CIS is implemented as a set of replicated physical units (CIS replicas) according to fault and intrusion tolerance needs. Likewise, CIS are interconnected with intrusion-tolerant protocols, in order to cooperate to implement the desired services.

3 CIS Overview

CIS is the substation gateway interfacing a protected LAN with the WAN, as shown in Fig. 1. In order to be intrusion-tolerant, the CIS is replicated (with diversity) in n machines and follows its specification as long as at most f of

[6] Some examples are the administrative clients and the servers LANs, the operational (SCADA) clients and servers LANs, the engineering clients and servers LANs, the Public Switched Telephone Network (PSTN) modem access LANs, the Internet and extranet access LANs, etc.

these machines are attacked and behave maliciously, both toward other replicas and toward the station computers in the protected LAN. Both the incoming and outgoing traffic is managed by "Traffic Replication Devices" that behave like Ethernet hubs: when they receive a packet from a port, they broadcast it to all the other ports. This way, the traffic received by the CIS from the WAN is spread to all the replicas, and the traffic generated by each replica is spread to all the other replicas and to the protected LAN.

The CIS is implemented using an hybrid architecture, so it is composed by two parts: the payload and the wormhole [8]. The payload is an asynchronous system where applications and protocols are executed; the wormhole is a secure and synchronous system providing services to the payload part through a well-defined interface. The wormhole part of each replica (local wormhole) is connected to the other local wormholes through a synchronous and secure control channel, isolated from other networks.

CIS intrusion tolerance is enhanced by rejuvenating CIS replicas through recoveries. In order to guarantee system availability despite the unavailability of recovering replicas, the number of replicas in the system is set to $n \geq 2f+1+k$, where k is the maximum number of replicas allowed to recover in parallel. This way, the system is able to tolerate at most f Byzantine replicas and recover k replicas simultaneously.

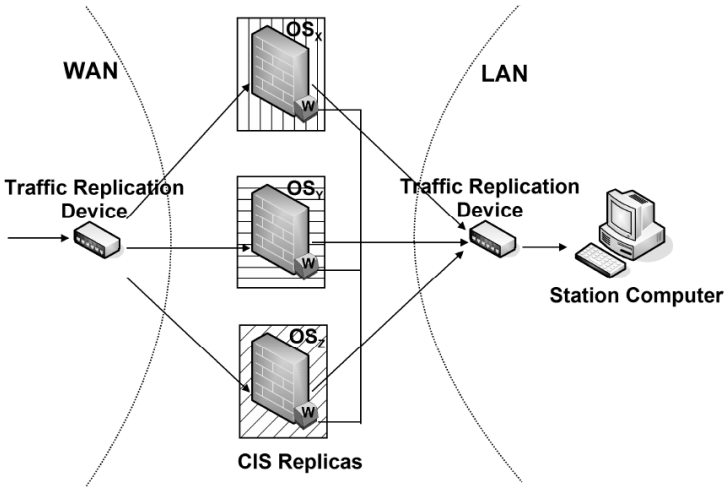


Fig. 1. CIS intrusion tolerant hybrid architecture

The CIS protection service, executed in each payload replica, verifies whether the incoming messages comply with the security policy (OrBAC[7] [9]), notifying their (positive) approval to its local wormhole. The wormhole collects message

[7] OrBAC (Organization Based Access Control)

approvals coming from the local wormholes; an incoming message m is signed by the wormhole if and only if the wormhole collects at least $f+1$ different approvals for m. Messages signed by the wormhole are considered valid messages and they are forwarded to their destination only by a distinguished payload replica, the leader (so there is no unnecessary traffic multiplication inside the LAN).

Each payload replica has to verify whether the leader forwards all the signed messages and has to check whether invalid messages are sent toward the LAN. The wormhole is in charge of both triggering the recoveries when necessary, ensuring that there is an amount of replicas to sustain system's correct operation, and managing the election of the new leader.

3.1 Fault Model and Assumptions

This Section describes [7] the fault model and the assumptions on which the fault model is based on. Station computers are assumed to only accept messages signed by the wormhole (a symmetric key K is shared between the station computer(s) and the CIS wormhole). The following faults are considered:

f1) The faults related to communication involve both the traffic replication devices and the communication channels among them and the replicas (except the control channel connecting local wormholes). Traffic replication devices can lose messages coming from a port or sometimes delay the traffic forwarding on some ports (for an unbounded time); traffic replication devices cannot generate spurious messages or alter messages. Communication channels can lose messages or unpredictably delay the traffic forwarding.
f2) A payload replica can be intruded, and hence can be affected by Byzantine faults.
f3) A local wormhole can only fail by crash; at most $f_c \leq f$ local wormholes are assumed to fail by crash. The crash of a local wormhole is detected by a perfect failure detector. When a local wormhole crashes, the corresponding payload is forced to crash together.
f4) Fault-independence is assumed for payload replicas, i.e., the probability of a replica being faulty is independent of the occurrence of faults in other replicas (this assumption can be substantiated in practice through the extensive use of several kinds of diversity [10]).
f5) The same attack on the same replica has always the same probability of success (this is a working assumption that could be relaxed in future work).
f6) Station computers cannot be compromised (it is the trusted network that we aim to protect, exactly in the sense of preventing it from being compromised).
f7) Replicas are correct after their recovery.
f8) The security policy verified by the CIS is assumed to be perfect; this means that a correct replica applies perfectly the policy verification and there are no policy inconsistencies between replicas (i.e. all correct replicas verify the same policy).

Given the set of faults just described, the corresponding failure modes for a payload replica are:

– Crash. The payload replica crashes because of the crash of the corresponding local wormhole (f3) or as the effect of an intrusion (f2).
– Omission. The payload replica is subjected to a transient omission because of communication problems (f1) or as the effect of an intrusion (f2). For example, a transient omission occurs when the leader payload is not forwarding a signed message because it never received it from the traffic replication device (f1).
– Invalid. The payload replica is failing by value as the effect of an intrusion (f2), e.g., it is sending illegal messages toward the LAN or it is flooding the WAN and the LAN aiming to delay the forwarding of legal messages.

For ease of modeling, we assume that a replica, as soon as it is successfully intruded, explicitly manifest failures (of any kind) and that a failure caused by an intrusion is permanent.

The system is unavailable if the number of correct working replicas is less than $f+1$ (so quorums cannot be reached) or if there are more than $f+1$ correct replicas, but the leader is omitting (so legal messages are not forwarded). The system fails if the number of invalid replicas is greater than f (the correctness of the system cannot be guaranteed) or if the necessary resources are unavailable for a fixed duration (CIS seeks perpetual operation).

3.2 The PRRW Strategy

We now explain the PRRW (Proactive-Reactive Recovery Wormhole) strategy that we are going to evaluate, laid down in [7]. The PRRW strategy manages the CIS replica recoveries using a mix of proactive and reactive recoveries, and it is characterized by the following parameters:

– The maximum time interval T_{P} (cycle or recovery period) between consecutive recoveries on the same replica (each replica is hence recovered at most after T_{P}).
– The worst case execution time T_{D} of a recovery.
– The maximum number k of replicas that may recover simultaneously.
– The maximum number f of simultaneously corrupted replicas that the system can tolerate.

The PRRW strategy is organized as shown in Fig. 2: time is divided in $\lceil n/k \rceil$ different time slots that are cyclically repeated. Each slot is divided in two tasks: task A and task R_i, with $i = 1, \ldots, \lceil n/k \rceil$.

Proactive (periodic) recoveries are executed during task R_i only; up to k replicas recover simultaneously in each task R_i, according to the replica index. Replica i, with $i = 1, \ldots, k$, are recovered in task R_1, replica i, with $i = k+1, \ldots, 2k$, are recovered in task R_2 and so on. Task R_i lasts for (at most) T_{D} and it is executed again after a period T_{P}.

Two types of reactive (a-periodic) recoveries can be triggered on replica i:

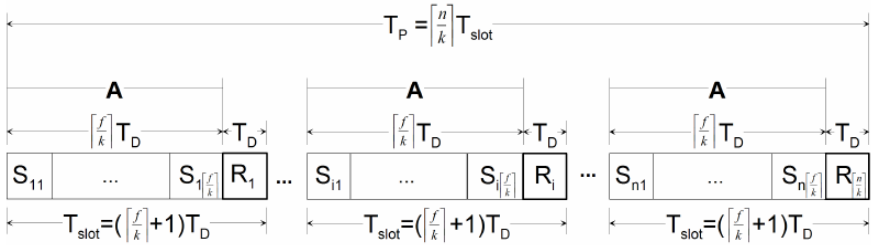


Fig. 2. The PRRW scheduling

1. "Immediate" reactive recovery, triggered if a quorum of $f+1$ accusations exists about i sending illegal messages; in this scenario replica i is "detected" of being compromised, because at least one correct replica detected that replica i is failed.
2. "Delayed" reactive recovery, triggered if a quorum of at least $f+1$ accusations exists about the current leader i, some about i sending illegal messages, other about i not forwarding a signed message (the signed messages was not forwarded for more then O_t times). In this scenario the leader replica i is "suspected" of being compromised, because at least one correct replica raised an accusation about leader replica i, but the wormhole is not able to identify which accuser replica is correct, so it is not able to identify which kind of accusation is correct about leader replica i.

"Immediate" reactive recoveries are immediately triggered on replica i as soon as the replica is detected of being compromised.

"Delayed" reactive recoveries are only triggered on the leader replica, are executed during task A and are coordinated with proactive recoveries. If no "immediate" reactive recovery is already triggered for replica i, the PRRW strategy finds the closest recovery sub-slot where the recovery of replica i does not endanger the availability of the CIS. If the found sub-slot is located in the slot where replica i will be proactively recovered, the "delayed" reactive recovery is not performed. Task A is divided into $\lceil f/k \rceil$ recovery sub-slots identified as S_{ij}; up to k replicas can be recovered simultaneously in each sub-slot. Task A lasts for (at most) $\lceil f/k \rceil T_D$.

Each slot lasts hence for up to $(\lceil f/k \rceil + 1) T_D$ with period T_P. After each R_i task has been executed once, each replica has been proactively recovered once.

A new leader is elected by the wormhole if the current leader is recovering or if the local wormhole of the current leader is detected to be crashed. The new leader is chosen as the (currently not crashed) replica more recently recovered by a proactive recovery.

4 PRRW Quantitative Analysis

This Section presents a quantitative analysis of the PRRW strategy. The relevant measures of interest are identified and the relevant parameters are described. The model representing the PRRW strategy is described and finally the results of the performed simulations are presented and discussed.

The quantitative analysis of the PRRW strategy aims to evaluate how effective is the trade-off between proactive and reactive recoveries. Proactive recoveries rejuvenate the replicas in predefined instants of time, without being based on any fault detection. This means that proactive recoveries treat all the faults, including also the latent and hidden ones, which cannot be treated in other way, but they recover also correct replicas, weakening the availability of the system. On the other side, reactive recoveries are triggered only on replicas detected or suspected of being faulty; replicas not detected or suspected of being faulty are never recovered, even if they are actually faulty, weakening the dependability of the system.

Recoveries determine a discontinuity in the CIS configuration caused by the temporary unavailability of the replicas subjected to a recovery. Therefore it is possible to represent the entire operational life split into different periods of deterministic duration called "phases". This feature allows a reconfiguration strategy belonging to the Multiple Phased System (MPS) class for which a modeling and evaluation methodology exist [11], supported by the DEEM tool [12]. Using DEEM, the model is split into two logically distinct sub-nets: the Phase Net (PhN) representing the schedule of the various phases, each one of deterministic duration, and the System Net (SN) representing the behavior of the system. Each net is made dependent on the other by marking-dependent predicates that modify transition rates, enabling conditions, reward rates etc. Reward measures are defined as Boolean expressions, functions of the net marking. Both the analytic [11] and simulation solutions [13] can be used in order to exercise the models; the measures of interest defined in our quantitative analysis were evaluated by simulation.

Different studies were performed on the modeled system varying several parameters; the relevant parameters are the following:

1. Mission time t.
2. Probability p_{I} of intrusion within a replica manifesting as a permanent invalid behavior; intrusions can manifest themselves as permanent omissions with probability $1-p_{\mathrm{I}}$. Parameter p_{I} impacts on the PRRW strategy because invalid and omission failures are treated in different ways.
3. Detection coverage c_{M} of malicious behavior of a replica. Parameter c_{M} impacts on the PRRW strategy because only detectable faults can trigger reactive recoveries.
4. Number n of replicas in the system.

The quantitative analysis aims to evaluate how these parameters impact on the measures of interest.

4.1 Measures of Interest

We are interested in measuring both the system failure probability $P_{\mathrm{F}}(t)$ and the system unavailability $P_{\mathrm{U}}(0,t)$ at time t.

The system fails at time t if one of the following conditions holds:

1. the number of invalid replicas gets over f;
2. the system is unavailable for an interval of time greater then T_{O}.

Let $P_{\mathrm{FI}}(t)$ be the probability of the system being failed at time t because of condition 1, given that it was correctly functioning at time $t = 0$. Let $P_{\mathrm{FO}}(t)$ be the probability of the system being failed at time t because of condition 2, given that it was correctly functioning at time $t = 0$. $P_{\mathrm{F}}(t)$ is defined as the probability of the system being failed at time t, given that it was not failed at time $t = 0$, and it is obtained as

$$P_{\mathrm{F}}(t) = P_{\mathrm{FI}}(t) + P_{\mathrm{FO}}(t).$$

The system is unavailable at time t if one of the following conditions holds:

1. the number of correct replicas is less than $f+1$ (quorums cannot be reached);
2. there are more than $f+1$ correct replicas, but the leader is omitting (legal messages are not forwarded).

Let $T_{\mathrm{U}}(0,t)$ be the total time the system is not failed but unavailable within $[0,t]$ because of one of the above conditions. Let $T_{\mathrm{A}}(0,t)$ be the total time the system is not failed within $[0,\ t]$. System unavailability, denoted by $P_{\mathrm{U}}(0,t)$, is defined as the probability of the system being unavailable within $T_{\mathrm{A}}(0,t)$, given that it was correctly working at time $t = 0$; system unavailability is obtained as

$$P_{\mathrm{U}}(0,\ t) = \frac{T_{\mathrm{U}}(0,\ t)}{T_{\mathrm{A}}(0,\ t)}.$$

4.2 The PRRW Model

The Phase Net (PhN). The phase net (Fig. 3) models the PRRW scheduling shown in Fig. 2. The deterministic transitions *TsubSlot* and *TRi* model the times to perform the tasks A and R_i, respectively. Place *Sij* contains a token during the task A (a-periodic recovery phase) and *Ri* contains a token during the task R_i (periodic recovery phase). The marking of *CountSubSlot* counts the number of the current recovery sub-slot (S_{ij}) within the current recovery slot. The marking of *CountSlot* counts the number of the current recovery slot within the current cycle. The marking of *CountWin* counts the number of the current cycle. The immediate transition *tNextSlot* fires when a periodic recovery slot ends, resetting the marking of *CountSubSlot* to 1. The immediate transition *tNextWin* fires when a new cycle is started, resetting the marking of *CountSlot* to 1. The immediate transitions of the phase net have priority less than the priorities of the immediate transitions of the system net.

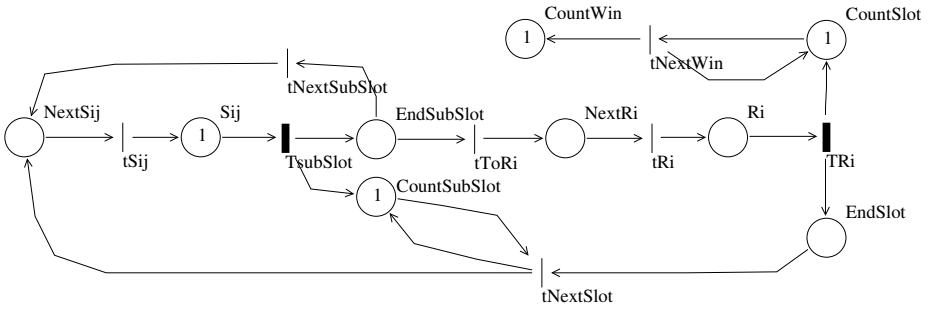


Fig. 3. The phase net of the PRRW model

The System Net (SN). The system net of the PRRW model is composed by $n \leq 6$ similar subnets (one subnet for each replica), a subnet to keep track of system failures and a subnet to model the initialization (the description of this last subnet is omitted without affecting the comprehension of the model).

Figure 4 shows the subnet modeling replica 1. The left part of the subnet models the replica failures, while the right part of the subnet models the replica recovery and leader election. Places which name ends with digit "1" model replica 1, while the other places (*Leader* and *kRec*) are shared by all the replicas.

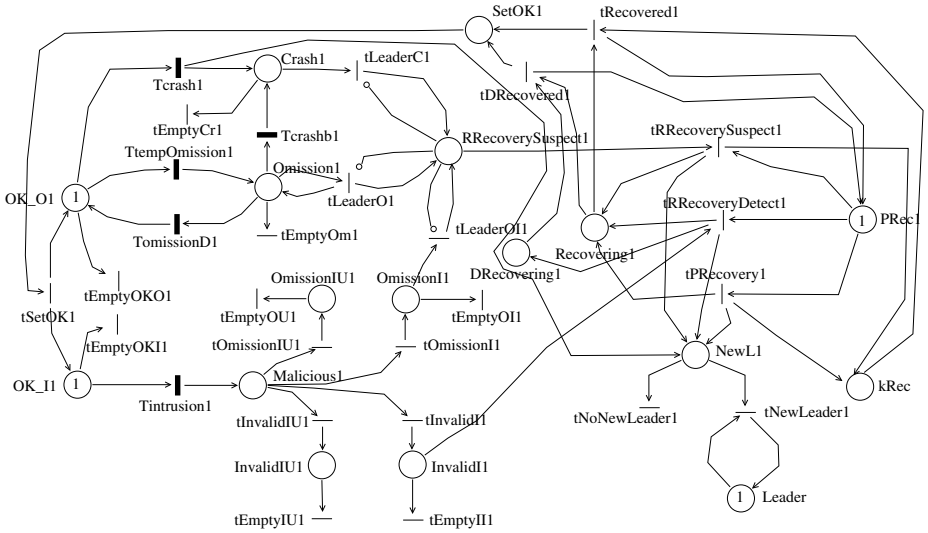


Fig. 4. The subnet of SN modeling replica 1

Replica failures are modeled as follows. As long as both *OK_O1* and *OK_I1* contain one token each, replica 1 is correctly working. One token in places *Crash1*

or *Omission1* represents the crash of the replica or an omissive behavior as a consequence of a transient omission, respectively. The exponential transitions *Tcrash1* and *Tcrashb1* represent the time to the crash with rate $\lambda_{\mathrm{I}}^{\mathrm{c}}$; when the replica crashes, place *OK_I1* is emptied (the replica cannot be no more intruded). *TtempOmission1* represents the time to a transient omission exponentially distributed with rate $\lambda_{\mathrm{I}}^{\mathrm{o}}$. A transient omission disappears after a time modeled by the exponential transition *TomissionD1* with rate λ^{eo}.

The exponential transition *Tintrusion1* represents the time to intrusion with rate $\lambda_{\mathrm{I}}^{\mathrm{a}}$; the effect of the intrusion is modeled by the following immediate transitions (enabled in the same marking) and the associated places:

- *TomissionIU1* for an undetectable omission failure, with probability $(1 - c_{\mathrm{M}})(1 - p_{\mathrm{I}})$,
- *TomissionI1* for a detectable omission failure, with probability $c_{\mathrm{M}}(1 - p_{\mathrm{I}})$,
- *TinvalidIU1* for an undetectable invalid failure, with probability $(1 - c_{\mathrm{M}})p_{\mathrm{I}}$,
- *TinvalidI1* for a detectable invalid failure, with probability $c_{\mathrm{M}}p_{\mathrm{I}}$,

where p_I and c_{M} are the probability of an intrusion manifesting as a permanent invalid behavior and the detection coverage of malicious behavior, respectively.

The replica recovery is modeled as follows. Place *PRec1* contains a token as long as replica 1 is not recovering, while place *Recovering1* contains one token as long as the replica is recovering. Place *DRecovering1* contains a token during a reactive recovery triggered by detections. Place *kRec* is used to count the number of replicas currently recovering. Place *RRecoverySuspect1* contains a token if a crash, an omission or a malicious omission occurs.

Recoveries are triggered by one of the following immediate transitions (ordered by increasing priorities): *tRRecoverySuspect1* (reactive recovery triggered by suspects), *tRRecoveryDetect1* (reactive recovery triggered by detections) or *tPRecovery1* (proactive recovery). The immediate transition *tRRecoverySuspect1* fires if a new a-periodic recovery sub-slot is starting (*NextSij* contains a token) and less than k replicas are recovering (*kRec* contains less than k tokens) and the replica is not going to be proactively recovered in the next periodic slot (the index of the replica is not in the interval $[(\mathit{Mark}(\mathit{CountSlot}) - 1)k + 1,\ \mathit{Mark}(\mathit{CountSlot})k]$). The immediate transition *tRRecoveryDetect1* fires if a new recovery sub-slot is starting (*NextSij* contains a token or NextRi contains a token). The immediate transition *tPRecovery1* fires if a periodic recovery slot is starting (*NextRi* contains a token) and less than k replicas are recovering (*kRec* contains less than k tokens) and the index of the replica is in the interval $[(\mathit{Mark}(\mathit{CountSlot}) - 1)k + 1,\ \mathit{Mark}(\mathit{CountSlot})k]$.

After the starting of a recovery of the replica, all the immediate transitions which name starts with *tEmpty* fire, emptying the following places: *OK_O1*, *OK_I1*, *Crash1*, *Omission1*, *InvalidIU1*, *InvalidI1*, *OmissionIU1* and *OmissionI1*. Immediate transitions *tRecovered1* or *tDRecovered1* fire when the current recovery ends, resetting the replica subnet.

The election of the leader replica is managed as follows. The marking of place *Leader* corresponds to the index of the current leader; when replica 1 either is going to be recovered or is crashed, one token is added in place *NewL1*.

tNewLeader1 fires if replica 1 is the current leader, triggering the mechanism of election of a new leader, otherwise *tNoNewLeader* fires. The arc from place *Leader* to place *tNewLeader1* has multiplicity equal to $Mark(Leader)$, while the arc from place *tNewLeader1* to place *Leader* has multiplicity equal to the index of the replica that will be elected as the new leader. The new leader should be the last (not crashed) replica proactively recovered, that is replica with index $j = ((n + (Mark(CountSlot) - 2)k) mod\, n) + k$. If replica j is currently crashed, the next attempt is made on replica $j - 1$, until a not crashed replica is found.

The subnet shown in Fig. 5 models the system failure. Place *OKSysN* contains a token as long as the system is not failed and it is not omitting (there are more than f correct replicas and the leader is not crashed or omitting). Place *OKSysO* contains a token when the system is not failed but it is omitting. Place *SysFailureI* contains a token when the system is failed because of invalid behavior (there are at least $f + 1$ invalid replicas). Place *SysFailureO* contains a token when the system is failed because the resource unavailability lasted for an unacceptable period of time represented by the exponential transition *TSysO* with rate $1/T_{\mathrm{O}}$.

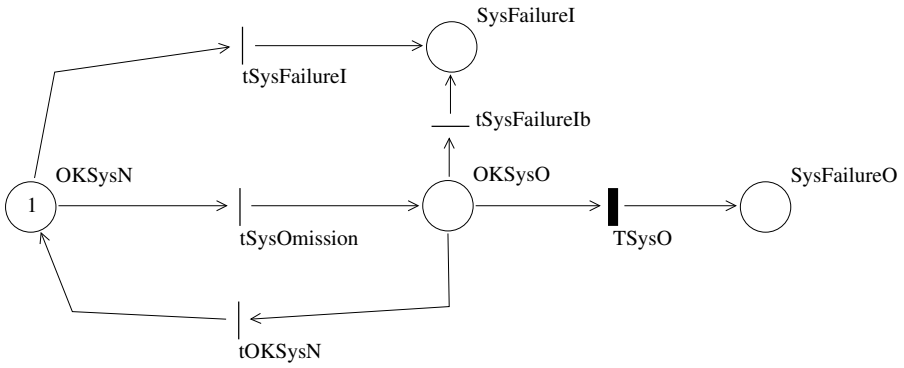


Fig. 5. The subnet of SN modeling the system failure

Different priorities are associated to the immediate transitions of SN, when no probabilistic choices are required. For example, all the immediate transitions of a replica i have priorities lower than those of replica j, if $i < j$.

Reward Structures. The evaluation of the measures of interest $P_{\mathrm{F}}(t)$ and $P_{\mathrm{U}}(0, t)$ involves specifying a performance (reward) variable and determining a reward structure for the performance variable, i.e., a reward structure which associates reward rates with state occupancies and reward impulses with state transitions [14]. System failure probability $P_{\mathrm{F}}(t)$ was evaluated in terms of an "instant of time" performance variable which is based on the following reward structure:

```
if (Mark(OKSysO)=0 and Mark(OKSysN)=0) then (1) else (0)
```

System unavailability $P_{\mathrm{U}}(0,t)$ was evaluated as $P_{\mathrm{U}}(0,\ t) = \frac{T_{\mathrm{U}}(0,\ t)}{T_{\mathrm{A}}(0,\ t)}$.

$T_{\mathrm{U}}(0,t)$ was evaluated defining an "interval of time" performance variable which reward structure is the following:

```
if (Mark(OKSysO)=1) then (1) else (0)
```

$T_{\mathrm{A}}(0,t)$ was evaluated defining an "interval of time" performance variable which reward structure is the following:

```
if (Mark(OKSysO)=1 or Mark(OKSysN)=1) then (1) else (0)
```

4.3 Model Evaluation and System Analysis

In this Section the results of the evaluation of the measures of interest are shown. The measures of interest were evaluated by simulation [13] with a confidence level of 95% and a half-length confidence interval of 1%.

All the model parameters and the default values used for the evaluations are shown in Table 1; the value for T_{D} was taken from [7]. The relevant parameters are:

1. Mission time t. This is the time during which the system is exercised since it starts to work. t varies in $[2628,\ 42048]$ sec.
2. Probability p_{I} of intrusion within a replica manifesting as a permanent invalid behavior. p_{I} varies in $[0,\ 1]$. If $p_{\mathrm{I}} = 0$ then all intrusions manifest as a permanent omissive behavior; in this case, only "delayed" reactive recoveries (on the leader replica) can be triggered. If $p_{\mathrm{I}} = 1$ then all intrusions manifest as a permanent invalid behavior; in this case, intrusions on each replica can only trigger "immediate" reactive recoveries.
3. Detection coverage c_{M} of malicious behavior of a replica. c_{M} is the probability of detecting an intruded replica, and hence the probability of reactively recovering an intruded replica. c_{M} varies in $[0,\ 1]$. If $c_{\mathrm{M}} = 0$ then no intrusions are detected; in this case, all intrusions are treated by proactive recoveries and reactive recoveries are only triggered by crash or communication omissions. If $c_{\mathrm{M}} = 1$ then all intrusions are detected and treated by reactive recoveries.
4. Number n of system replicas in the system, maximum number f of corrupted replicas tolerated by the system itself and maximum number k of system replicas recovering simultaneously, with $n = 2f + 1 + k$.

A first study was performed observing both system failure probability $P_{\mathrm{F}}(t)$ and system unavailability $P_{\mathrm{U}}(0,t)$ over mission time t for three different values of p_{I}.

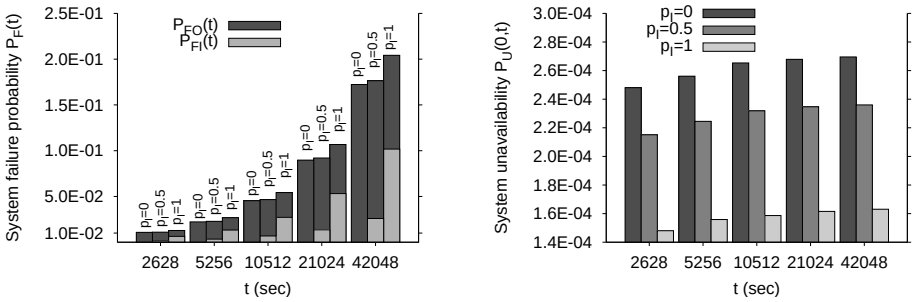
Table 1. Parameters and their default values

Name	Default Value	Meaning
t	2628	Mission time (sec)
n	4	Number of replicas in the system
k	1	Max number of replicas recovering simultaneously
f	1	Max number of corrupted replicas tolerated by the system
T_{D}	146	Time duration of a recovery operation (sec)
T_{O}	60	Duration of system omission before considering the system failed (sec)
λ_i^{c}	[1.9E-7, 3.8E-7]	Crash rate of replica i. Each replica has a diverse crash rate (from 1 per 60 days to 1 per 30 days)
λ_i^{o}	[1.9E-6, 3.8E-6]	Transient omission rate of replica i. Each replica has a diverse rate (from 1 per 6 days to 1 per 3 days)
λ^{eo}	3.3E-2	Omission duration rate of a replica. A transient omission lasts for 30 seconds (on average)
λ_i^{a}	[5.8E-5, 1.2E-5]	Successful attack (intrusion) rate of replica i. Each replica has a diverse rate (from 5 per day to 1 per day)
p_{I}	0.5	Probability of intrusion within a replica manifesting as a permanent invalid behavior (if $p_{\mathrm{I}} = 0$ all intrusions manifest as permanent omissions)
c_{M}	0.7	Probability of detecting malicious behavior of a replica

Figure 6(a) shows how $P_{\mathrm{FI}}(t)$ and $P_{\mathrm{FO}}(t)$ change over mission time t, with $P_{\mathrm{F}}(t) = P_{\mathrm{FI}}(t) + P_{\mathrm{FO}}(t)$. $P_{\mathrm{F}}(t)$ increases exponentially over time for all the values of p_{I}. $P_{\mathrm{F}}(t)$ behaves in fact like a geometric random variable for the following reasons. System failure probability during each recovery period (cycle) is not null; after each cycle the system is rejuvenated, so we can assume that the system failure probability during the next cycle is the same as the previous one. So system failure probability $P_{\mathrm{F}}(t)$ cumulates over the recovery periods as a geometric random variable. The values of $P_{\mathrm{F}}(t)$ are over 0.01 because of the values assigned to the system parameters. As p_{I} varies from 0 to 1, $P_{\mathrm{F}}(t)$ increases of about 30% for low values of t and increases of about 17% for high values of t. For $p_{\mathrm{I}} = 0$, $p_{\mathrm{I}} = 0.5$ and $p_{\mathrm{I}} = 1$ the value of $P_{\mathrm{FI}}(t)$ is about 0%, 17% and 50% of the value of $P_{\mathrm{F}}(t)$, respectively, independently on the values of t.

If $p_{\mathrm{I}} = 0$ then $P_{\mathrm{FI}}(t) = 0$, because there is no invalid behavior, and hence $P_{\mathrm{F}}(t) = P_{\mathrm{FO}}(t)$. As p_{I} varies from 0 to 1, $P_{\mathrm{FO}}(t)$ changes from 100% of $P_{\mathrm{F}}(t)$ to 50% of $P_{\mathrm{F}}(t)$; the number of intrusions does not change, but the effect of intrusions changes. In fact, the value of $P_{\mathrm{FO}}(t)$ depends on the time during which replicas are unavailable, which for $p_{\mathrm{I}} = 0$ is given by the sum of the following durations:

- the time spent waiting for a "delayed" reactive recovery of the omissive leader;
- the time spent during the recovery on the omissive leader;



(a) System failure probability $P_{\mathrm{F}}(t)$ (b) System unavailability $P_{\mathrm{U}}(0,t)$

Fig. 6. System failure probability $P_{\mathrm{F}}(t)$ and system unavailability $P_{\mathrm{U}}(0,t)$ over mission time t for different values of p_{I}

- the time spent waiting for proactive recoveries of (not leader) omissive replicas;
- the time spent for proactive recoveries (not varying for the different values of p_{I}).

If $p_{\mathrm{I}} = 1$ then the time during which replicas are unavailable is given by the sum of the following durations:

- the time spent during "immediate" reactive recoveries on replicas detected as intruded; the number of these recoveries is about n times the number of "delayed" reactive recoveries performed for $p_{\mathrm{I}} = 0$;
- the time spent for proactive recoveries.

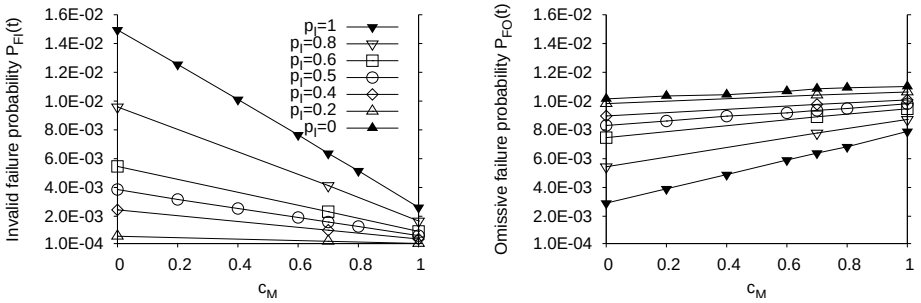
Therefore, the value of $P_{\mathrm{FO}}(t)$ for $p_{\mathrm{I}} = 1$ mainly represents the impact of recoveries (both proactive and reactive) on $P_{\mathrm{F}}(t)$ (crashes and transient omissions are still present, but have lower rates than intrusions). The value of $P_{\mathrm{FO}}(t)$ for $p_{\mathrm{I}} = 1$ shows that the impact of recoveries on $P_{\mathrm{F}}(t)$ is high (about 50%).

Figure 6(b) shows how $P_{\mathrm{U}}(0,t)$ changes over mission time t. $P_{\mathrm{U}}(0,t)$ increases over time for all the values of p_{I}. For $p_{\mathrm{I}} = 0.5$ and $p_{\mathrm{I}} = 1$ the value of $P_{\mathrm{U}}(0,t)$ is about 87% and 60% of the value of $P_{\mathrm{U}}(0,t)$ for $p_{\mathrm{I}} = 0$, respectively, independently on the values of t.

The trend of $P_{\mathrm{U}}(0,t)$ for varying p_{I} is similar to the trend of $P_{\mathrm{FO}}(t)$ shown in Fig. 6(a); for $p_{\mathrm{I}} = 1$ the value of $P_{\mathrm{U}}(0,t)$ is mainly due to the recoveries, for $p_{\mathrm{I}} = 1$ and $p_{\mathrm{I}} = 0.5$ the value of $P_{\mathrm{U}}(0,t)$ is influenced by the fact that the number of recoveries decreases but the number of omission increases.

Another study was devoted to evaluate both system failure probability $P_{\mathrm{F}}(t)$ and system unavailability $P_{\mathrm{U}}(0,t)$, varying both the detection coverage c_{M} and the probability p_{I} of intrusions manifesting as invalid behavior. This study shows how reactive recoveries improve the measures of interest with regard to treating intrusions with proactive recoveries only.

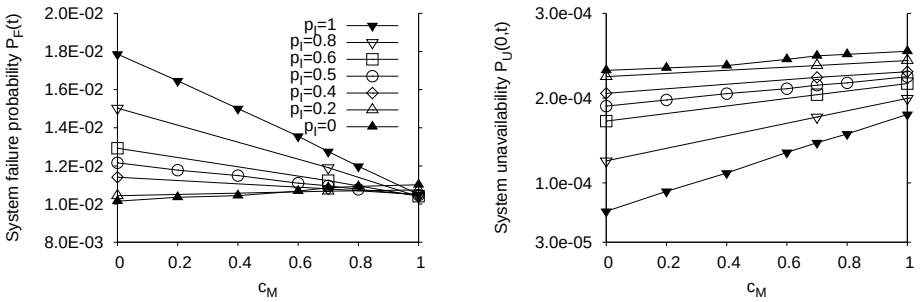
Figures 7(a) and 7(b) show how $P_{\mathrm{FI}}(t)$ and $P_{\mathrm{FO}}(t)$, respectively, change over detection coverage c_{M} for different values of p_{I}; in order to make easier their comparison, the same scale for the y-axis is used. $P_{\mathrm{FI}}(t)$ decreases as c_{M} increases from 0 to 1 for all the values of p_{I}. $P_{\mathrm{FI}}(t)$ takes the largest values for $p_{\mathrm{I}} = 1$ and the lowest values for $p_{\mathrm{I}} = 0$. If $p_{\mathrm{I}} = 0$ then the values of $P_{\mathrm{FI}}(t)$ for different values of c_{M} are 0 and are not shown in Fig. 7(a). $P_{\mathrm{FI}}(t)$ takes the smallest values for $p_{\mathrm{I}} = 0.2$ and is almost constant. The curve corresponding to $p_{\mathrm{I}} = 1$ decreases quicker than the other curves (it decreases for about one order of magnitude) as c_{M} increases. $P_{\mathrm{FO}}(t)$ shows an opposite behavior with respect to $P_{\mathrm{FI}}(t)$: it increases as c_{M} increases from 0 to 1. $P_{\mathrm{FO}}(t)$ takes the largest values for $p_{\mathrm{I}} = 0$ and the lowest values for $p_{\mathrm{I}} = 1$. The curve corresponding to $p_{\mathrm{I}} = 1$ increases quicker than the other curves (it increases for about 2.5 times); the curve corresponding to $p_{\mathrm{I}} = 0$ is almost constant. The largest variations in the values of $P_{\mathrm{FI}}(t)$ and $P_{\mathrm{FO}}(t)$ for varying c_{M} occur for $p_{\mathrm{I}} = 1$.



(a) Invalid failure probability $P_{\mathrm{FI}}(t)$ (b) Omissive failure probability $P_{\mathrm{FO}}(t)$

Fig. 7. Impact of detection coverage c_{M} on both $P_{\mathrm{FI}}(t)$ and $P_{\mathrm{FO}}(t)$ for different values of p_{I}

The values of $P_{\mathrm{FI}}(t)$ and $P_{\mathrm{FO}}(t)$ for $c_{\mathrm{M}} = 0$ correspond to the system configuration in which all the intrusions are treated only by proactive recoveries. The difference between the values of $P_{\mathrm{FI}}(t)$ (and $P_{\mathrm{FO}}(t)$) for $c_{\mathrm{M}} = 0$ and $c_{\mathrm{M}} = 1$ is due to the effect of treating all the intrusions by reactive recoveries. Increasing c_{M} there are two opposite effects with respect to $P_{\mathrm{FI}}(t)$ and $P_{\mathrm{FO}}(t)$: $P_{\mathrm{FI}}(t)$ decreases, because invalid replicas reactively recovered are no longer weakening the system; $P_{\mathrm{FO}}(t)$ increases, because replicas, while recovering, do not contribute to system operation. The overall effect, shown in Fig. 8(a), is that, when most of the intrusions behave as invalid ($p_{\mathrm{I}} \geq 0.4$), system failure probability $P_{\mathrm{F}}(t)$ decreases as detection coverage c_{M} increases. On the contrary, when most of the intrusions behave as omissions ($p_{\mathrm{I}} < 0.4$), the impact of c_{M} on $P_{\mathrm{F}}(t)$ is negligible. This stresses that, in order to improve the value of $P_{\mathrm{F}}(t)$, it is useful to trigger reactive recoveries and hence to set the value for c_{M} as higher as possible.



(a) System failure probability $P_{\mathrm{F}}(t)$ (b) System unavailability $P_{\mathrm{U}}(0,t)$

Fig. 8. Impact of detection coverage c_{M} on system failure probability $P_{\mathrm{F}}(t)$ and system unavailability $P_{\mathrm{U}}(0,t)$ for different values of p_{I}

Figure 8(b) shows how system unavailability $P_{\mathrm{U}}(0,t)$ changes over detection coverage c_{M} for different values of p_{I}. The trend of $P_{\mathrm{U}}(0,t)$ for varying c_{M} is similar to the trend of $P_{\mathrm{FO}}(t)$ shown in Fig. 7(b). $P_{\mathrm{U}}(0,t)$ increases as c_{M} increases from 0 to 1 for all values of p_{I}. $P_{\mathrm{U}}(0,t)$ takes the largest values for $p_{\mathrm{I}} = 0$ and the lowest values for $p_{\mathrm{I}} = 1$. If $p_{\mathrm{I}} = 0$, $P_{\mathrm{U}}(0,t)$ is almost not influenced by changing the detection coverage, while increasing p_{I} the influence of c_{M} becomes more evident (almost an order of magnitude for $p_{\mathrm{I}} = 1$).

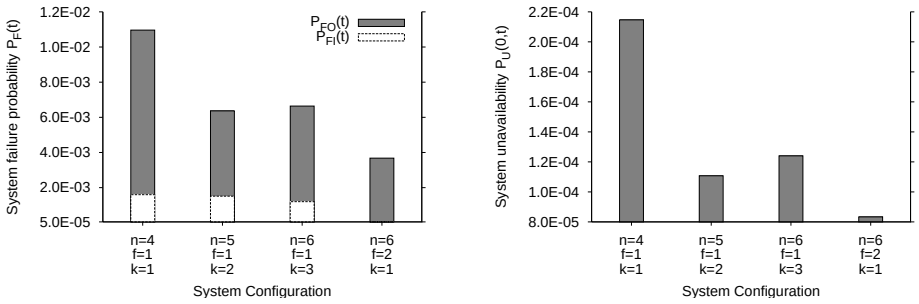
It turns out that $P_{\mathrm{U}}(0,t)$ is negatively affected by a larger value for c_{M}, because the larger is the detection coverage, the more reactive recoveries are triggered; the above trend is more evident as the probability p_{I} increases, because recoveries triggered by invalid behavior involve all replicas, not only the leader.

The results of this study show that increasing the detection coverage of intrusions has conflicting effects on system failure probability $P_{\mathrm{F}}(t)$ and system unavailability $P_{\mathrm{U}}(0,t)$: as c_{M} increases, $P_{\mathrm{F}}(t)$ improves and $P_{\mathrm{U}}(0,t)$ gets worsen; the impact of this effect depends on the behavior of the (invalid or omissive) intrusions, i.e. on the value of the parameter p_{I}. Since a low $P_{\mathrm{F}}(t)$ and a low $P_{\mathrm{U}}(0,t)$ are conflicting goals, the proper tuning of c_{M} entails defining their relative importance. Thus, if $P_{\mathrm{F}}(t)$ has to be optimized, high values of c_{M} are required, while low values of c_{M} optimize $P_{\mathrm{U}}(0,t)$. More generally, parameters for the CIS system can be tuned once the system designer has given constraints on the desired behavior of the system, e.g., $P_{\mathrm{F}}(t)$ must be optimized while $P_{\mathrm{U}}(0,t)$ must take values lower than a given threshold.

The last study performed aimed to evaluate the impact of the number of replicas on both system failure probability $P_{\mathrm{F}}(t)$ and system unavailability $P_{\mathrm{U}}(0,t)$. When dealing with the number of replicas in the system, three parameters are relevant: n, the overall number of replicas in the system, f, the maximum number of corrupted replicas tolerated by the system and k, the maximum number of replicas simultaneously recovering without endangering the availability of the system, with $n = 2f + 1 + k$. The following system configurations were evaluated:

1. $n = 4, f = 1, k = 1$
2. $n = 5, f = 1, k = 2$
3. $n = 6, f = 1, k = 3$
4. $n = 6, f = 2, k = 1$

Figures 9(a) and 9(b) show system failure probability $P_{\mathrm{F}}(t)$ (decomposed in $P_{\mathrm{FI}}(t)$ and $P_{\mathrm{FO}}(t)$) and system unavailability $P_{\mathrm{U}}(0,t)$ for the system configurations described above. $P_{\mathrm{FI}}(t)$ decreases as n (and k) increases. The trend of $P_{\mathrm{F}}(t)$ is mainly due to the trend of $P_{\mathrm{FO}}(t)$. For the same value of $n = 6$, the higher is f and the lower is $P_{\mathrm{F}}(t)$ (both $P_{\mathrm{FI}}(t)$ and $P_{\mathrm{FO}}(t)$), although k is lower. $P_{\mathrm{FI}}(t)$ is lower because of the intrusion tolerance scheme is more robust ($f = 2$); $P_{\mathrm{FO}}(t)$ is lower because the frequency of proactive recoveries is lower ($k = 1$). The trend of $P_{\mathrm{U}}(0,t)$ is the same of $P_{\mathrm{F}}(t)$.



(a) System failure probability $P_{\mathrm{F}}(t)$ (b) System unavailability $P_{\mathrm{U}}(0,t)$

Fig. 9. System failure probability $P_{\mathrm{F}}(t)$ and system unavailability $P_{\mathrm{U}}(0,t)$ for different system configurations at mission time $t = 2628\,\mathrm{sec}$

We suppose that the increment of the value of $P_{\mathrm{FO}}(t)$ varying from configuration 2 to 3 is due to the combined effect of a larger number of failures (n varies from 5 to 6, but $f = 1$) and a higher frequency for proactive recoveries (k varies from 2 to 3). It turns out that for the setting used (as shown in Table 1) the lower values for $P_{\mathrm{F}}(t)$ and $P_{\mathrm{U}}(0,t)$ are obtained for the system configuration 4, i.e., for higher values of f, independently of k.

4.4 Discussion about the PRRW Strategy

The CIS intrusion tolerance is currently obtained through a recovery strategy (PRRW) based on a combination of proactive and reactive recoveries. The use of both proactive and reactive recoveries shows to be effective since the two techniques possess complementary characteristics.

Proactive recoveries periodically rejuvenate all the replicas, without any need of fault detection mechanisms (also latent/hidden faults are treated). The period

of the proactive recoveries defines a bounded temporal window (between two recoveries of the same replica) which represents a time limit for an attack attempt to be successful. In fact, this is the time an attacker has for conquering a majority of the replicas and thus for taking the control of the entire CIS. On the other side, being an "unconditional" recovery, the proactive recovery is applied also to correct replicas which become unavailable for the time necessary to perform the recovery. Moreover, if only proactive recovery is used in a system, a replica hit by a fault will be unavailable until the end of its next proactive recovery.

On the contrary, a reactive recovery is triggered only when a fault of a replica is detected, so its effectiveness depends on the assumed fault model and on the coverage of the detection mechanism used (latent/hidden faults are not treated). As shown in Fig. 8(a), reactive recoveries of the faulty replicas contribute to decrease system failure probability; they are in fact performed as soon as possible, however within the duration of $\lceil f/k \rceil T_{\mathrm{D}}$, without waiting the next periodic recovery on the same replica. In this way, the recovery and the rejuvenation of a faulty replica is anticipated with respect to its next proactive recovery, so the (faulty) replica becomes active and correct earlier.

This behavior apparently suggests that the more reactive recoveries are performed, the worse is system availability, as it appears evidently in Fig. 8(b) for $p_{\mathrm{I}} = 1$. In this case, all the intrusions manifest as invalid behavior and all the detected intrusions trigger a reactive recovery. In reality, what happens is that the system ability to survive gets increased, whereas for low values of the coverage (thus less reactive recoveries) the system fails as soon as replicas get affected by faults.

The PRRW strategy, as our analysis reveals, makes a significant difference in the way omission and invalid behaviors are treated. This is made evident by observing all the curves at varying values for p_{I}. Actually, invalid behaviors are detected with coverage c_{M} and trigger a reactive recovery, whereas omissive behaviors are essentially not detected: only the omission of the leader is detected and triggers some action, the omissions of the followers are removed only with the proactive recovery. Increasing the capability to detect (and quickly react) to omissive behaviors is a way to improve the overall fault tolerance strategy.

5 Direction for Improvements/Refinements

This Section identifies the directions for refining and improving the recovery strategy. An extended fault model is introduced and some modifications to the recovery schemes are presented.

5.1 New Extended Fault Model

The reactive recovery of the PRRW strategy is based on distinguishing and detecting a limited set of faults in replicas, amongst those possible to occur. Obviously, the remainder faults are treated, thanks to the strategy of proactive recoveries. We analyze this situation, under the light of the evaluation just

performed, and enumerate a possible set of additional faults to be taken into account, in the sense of improving both system dependability and availability.

In the PRRW strategy, the correct replicas detect the following faults:

- Leader Benign Fault (LBF): The faulty leader omits to send a signed message to the LAN. A correct replica will suspect the leader to be "silent" after O_t consecutive leader omissions on the same signed message.
- Replica Malicious Fault (RMF): The faulty replica (being it either the leader or a follower) sends an unsigned message to the LAN; a correct replica will immediately detect the faulty replica to be a "malicious sender".

It comes out that the PRRW schema takes into account both omissive and malicious faults in the leader replica, but only malicious faults in the follower replicas. The idea is that if a follower is going to have an omissive behavior, the problem will be eventually treated either by the proactive recovery or by the election of the replica as a leader (the replica will be extensively monitored in this case). In both cases, the negative effects of the faults will be eventually eliminated.

An additional set of faults might be considered by the current reactive recovery mechanisms, since detecting such faults and treating them using reactive recoveries would improve both dependability and availability of the system. These faults are listed below:

- Malicious Approval (MA): A faulty replica approves an illegal message; the faulty replica is intruded, because all correct replicas verify the same security policy.
- Omitted Approval (OA): A faulty replica omits to approve a legal message; the omission could be caused by communication problems (the replica never received the legal message), but it could be also the effect of an intrusion.
- Malicious Suspect (MS): A faulty replica signals the wormhole an accusation about a correct replica; the faulty replica is intruded, because a correct replica does not show any incorrect behavior.
- Omitted Suspect (OS): A faulty replica does not signal the wormhole any accusation about a faulty replica; the omission could be caused by communication problems (the replica never received the legal message), but it could also be the effect of an intrusion.

In the MA and MS cases, the faulty replica is intruded, so it needs to be recovered as soon as possible; if the faulty replica is not detected as such, it is still considered correct. In the OA and OS cases, faults could be caused either by communication omissions (no recovery is useful to solve the problem) or as an effect of intrusions manifesting as omissive behavior (a recovery could solve the problem). Devising the adequate mechanisms for faithful detection is a subject of further study, but we underline possible avenues in the next section.

5.2 Architectural Modifications for the Detection of the Extended set of Faults

This Section describes the architecture modifications necessary to detect the faults described in Sect. 5.1 and trigger the reactive recoveries. In order to perform the detection of the above faults it is necessary to allow each payload replica to be informed about all the approval results and manifested suspects taken by all the other payload replicas.

A shared virtual memory (SVM) mechanism [15, 16] can be implemented as a reliable repository where each replica posts all its approval results and suspects; a majority of correct replicas is thus able to identify which replicas took the wrong approval decisions (if any) or manifested the wrong suspect (if any).

Approval results are stored for each incoming message as a data structure containing i) an identification for the incoming message m, ii) the approval decisions collected from all the replicas about m, iii) the final vote given by the wormhole about m. Suspects are stored as a data structure containing the suspecter(s), the suspected and the kind of suspect. Information is stored in the shared virtual memory, using it as a circular buffer in order to make room for newer information; therefore the SVM is used as a queue of dimension q. If the information to be broadcasted should be too heavy to be managed through the wormhole, some form of "compression" can be found.

Each message is identified using its MAC. Each approval decision is stored in an array of n elements, where the i-th element represents approval result of replica i about message m:

- "ACCEPT": replica i approves m;
- "REJECT": replica i does not approve m;
- "null": no approval information still received from replica i about m;
- "recovering": replica i is currently recovering.

The final vote can be one of the following: "LEGAL", "ILLEGAL" and "VOTING".

The follower payload behavior is monitored as follows. When message m comes from the WAN, each replica decides whether approving it or not, posting the final decision in the SVM. Not all the replicas will receive m in the same instant, and each replica will need some time in order to take the approval decision and post it in the repository, but a certain number of approval results about m will be available in the SVM at worst within T_{vote} time after the first post. Replicas that did not take any approval result till that moment and that were not recovering (those corresponding to the "null" array elements) will be suspected of omission (they could not have received m because of communication faults or they could have omit maliciously). Given the final vote about m, all the correct replicas (i.e. all the replicas which approval result is in agreement with the final vote) will be able to identify all the faulty ones (i.e. all the replicas which approval result is in disagreement with the final vote) and suspect them as malicious faulty replicas.

6 Concluding Remarks

This chapter analyzed the basic components of the CIS (CRUTIAL Information Switch) architecture proposed in the framework of the European Project CRUTIAL, where an infrastructure architecture seen as a WAN of LANs has been proposed. LANs confine existing sub-systems, protected by special interconnection and filtering devices (CIS); a set of CIS can collectively ensure that the computers controlling the physical process correctly exchange information despite accidents and malicious attacks.

We identified two dependability and availability measures of interest. We constructed a model of the the CIS recovery scheme, called PRRW, and we performed a preliminary analysis of the quantitative behavior of the PRRW. We analyzed and discussed the impact of some relevant parameters as the detection coverage, the intrusions and the number of CIS replicas, on the measures of interest, aiming to evaluate how effective is the trade-off between proactive and reactive recoveries. In particular, we have shown that increasing the detection coverage of intrusions has conflicting effects on both dependability and availability measures, and that these effects depend also on the behavior of invalid or omissive intrusions. The directions for refining and improving the recovery strategy were proposed.

Further studies are envisioned mainly in the following directions. We will deeply analyze the impact on the measures of interest of some PRRW parameters not yet investigated, like, for example, the false positive in intrusion detection, the threshold of duration of system omission before considering the system failed. We will deeply analyze the impact of the number of replicas (parameters n, f and k) and the duration of the recovery period (i.e. the frequency of proactive recoveries). We will analyze alternative recovery policies, for example a recovery strategy where proactive recoveries are anticipated to the first available slot (slots where reactive recoveries are not requested).

Acknowledgments

This work has been partially supported by the European Community through the IST Projects CRUTIAL (Contract n. 027513).

References

1. Madani, V., Novosel, D.: Getting a grip on the grid. Spectrum, IEEE **42** (2005) 42–47
2. Dawson, R., Boyd, C., Dawson, E., González Nieto, J.: SKMA: a key management architecture for SCADA systems. In: ACSW Frontiers '06: Proceedings of the 2006 Australasian workshops on Grid computing and e-research, Darlinghurst, Australia, Australian Computer Society, Inc. (2006) 183–192
3. Wilson, C.: Terrorist capabilities for cyber-attack. In: Dunn and V. Mauer, editors, Int. CIIP Handbook volume II, CSS, ETH Zurich (2006) 69–88

4. Gordon, L., Loeb, M., Lucyshyn, W., Richardson, R.: 2006 CSI/FBI computer crime and security survey (2006)
5. Veríssimo, P., Neves, N., Correia, M.: CRUTIAL: The blueprint of a reference critical information infrastructure architecture. In: 1st International Workshop on Critical Information Infrastructures @ ISC06. (2006)
6. Sousa, P., Neves, N., Lopes, A., Veríssimo, P.: On the resilience of intrusion-tolerant distributed systems. DI/FCUL TR 6–14, Department of Informatics, University of Lisbon (2006)
7. Sousa, P., Bessani, A., Correia, M., Neves, N., Veríssimo, P.: Resilient intrusion tolerance through proactive and reactive recovery. In: 13th IEEE Pacific Rim Dependable Computing conference. (2007)
8. Veríssimo, P.: Travelling through wormholes: a new look at distributed systems models. SIGACT News **37** (2006) 66–81
9. Abou El Kalam, A., El Baida, R., Balbiani, P., Benferhat, S., Cuppens, F., Deswarte, Y., Miège, A., Saurel, C., Trouessin, G.: Organization based access control. In: 4th IEEE Int. Workshop on Policies for Distributed Systems and Networks. (2003)
10. Obelheiro, R., Bessani, A., Lung, L., Correia, M.: How practical are intrusion-tolerant distributed systems? DI/FCUL TR 06–15, Department of Informatics, University of Lisbon (2006)
11. Mura, I., Bondavalli, A.: Markov regenerative stochastic Petri nets to model and evaluate the dependability of phased missions. IEEE Transactions on Computers **50** (2001) 1337–1351
12. Bondavalli, A., Mura, I., Chiaradonna, S., Filippini, R., Poli, S., Sandrini, F.: DEEM: a tool for the dependability modeling and evaluation of multiple phased systems. In: DSN-2000 IEEE Int. Conference on Dependable Systems and Networks (FTCS-30 and DCCA-8). (2000) 231–236
13. Moretto, M.: Progettazione, realizzazione ed utilizzo di un generatore di simulatori per sistemi a fasi multiple. Master's thesis, Università degli Studi di Pisa (2004)
14. Sanders, W., Meyer, J.: A unified approach for specifying measures of performance, dependability and performability. In Avizienis, A., Laprie, J., eds.: Dependable Computing for Critical Applications, Vol. 4 of Dependable Computing and Fault-Tolerant Systems. Springer Verlag (1991) 215–237
15. Nitzberg, B., Lo, V.: Distributed shared memory: a survey of issues and algorithms. Computer **24** (1991) 52–60
16. Morin, C., Puaut, I.: A survey of recoverable distributed shared virtual memory systems. Parallel and Distributed Systems, IEEE Transactions on **8** (1997) 959–969