An Agent Decision Model for an Adaptive Supervision of Distributed Systems

Cédric Herpson*, Amal El Fallah Seghrouchni, Vincent Corruble
LIP6, UPMC, équipe SMA
4 place Jussieu, 75005, Paris, France
{cedric.herpson,amal.elfallah,vincent.corruble}@lip6.fr

Abstract. The advent of complex and physically distributed systems and the need to minimize the down-time of services and production processes call for more efficient supervision systems. Traditionally centralized, the anytime supervision of such systems is challenged when communications between the supervision and the supervised systems become either slow, disrupted or too costly. In this paper, we propose an agent decision model that allows a multi-agent supervision system to dynamically adapt itself to the state of the communications by means of distributing the diagnosis and repair process. Experiments on a simulator for distributed systems using an industrial dataset show that our proposal does lead to an anytime and adaptive supervision of distributed systems where a short response time prevails over a limited repair extra-cost.

Keywords: Multi-Agent, Supervision, Anytime, Diagnosis, Repair, Distributed Systems

1 Introduction

Complex and distributed systems such as ones found in services, production, large infrastructures and organizations, require a layer of supervision that can automatically deal with the important tasks of monitoring, diagnosis, and repair. Indeed, as the complexity of systems increases, humans can no longer process the flow of information arriving at each instant. So, the improvement of system effectiveness requires the delegation of more tasks to the supervision system, and more automation of its decisions/actions. This requirement has lead to the (re)birth of a research community around the notions of autonomic computing [KC03] and self-* systems [ST09]. Our work lies within this context.

Within the Dem@tFactory project, our objective is thus to improve the supervision of an existing digitizing chain distributed over 5 sites (3 in France, 1 in Madagascar and 1 in Mauritius). Different faults – single or multiple – can

---

1 Project of the French R&D initiative Cap Digital federating 4 industrials and 3 laboratories, the Dem@tFactory project aims to improve the functioning of a digitizing chain of heterogeneous documents
occur and alter or prevent the processing of the documents (e.g. a scanner quit working, a disruption of the connection between different sites can halt or corrupt data transfer, an OCR software may be poorly set, etc.).

Currently, and as it is commonly found in the industry, the supervision of the digitization process is centralized. However, the centralized approach does not perform well in a distributed and asynchronous context. Indeed, communication malfunctions between the supervision system and the geographically distributed regions of the supervised system delay the repair and do not allow the digitization chain to quickly return to normality, even though a number of malfunctions have predefined repair procedures. The unbounded – and sometimes unnecessary – communication time reduces the effectiveness of the supervision and therefore increases the down-time of the supervised system.

Many distributed approaches are proposed in the literature to address this difficulty. However, whether work is from the diagnosis and control communities [Mou08,CL08,LTS09,JKK05,QLK05,QLK06] or from the multi-agent domain [WLA04,RTBW02,HLV00,LKK09,KT00], they do not integrate the repair phase. This last part is generally left to the responsibility of humans, which require to centralize the result of the diagnosis and lead the supervision system to waste the time saved by the distribution of the diagnosis phase. To our knowledge, the work of Nejdl et al. [NW94,FN92] is the only ones who have addressed the distribution of both the diagnosis and repair phases. However, their work makes the assumptions that communication links are reliable and that messages can be exchanged between agents at no cost. In a real situation these hypotheses are too restrictive. Indeed, to not consider the communication state may render the supervision system ineffective or inoperable.

In the context of the Dem@tFactory project, our goal is to propose a supervision mechanism that can guarantee to be effective even in a context of unreliable communications. Our proposal stands on a multi-agent architecture where each supervision agent handles both diagnosis and repair on a given location. Our architecture comes within the scope of model-free approaches with spatially distributed knowledge [RTBW02]. We have identified three key points for obtaining an anytime supervision architecture: (1) the agent decision model, (2) the inference mechanism for diagnosis computation, and (3) the maintenance of a consistent view of the system state by the multi-agent system.

1. **Decision model** In a distributed environment, having an anytime supervision system requires that each agent is able to make a decision at any time depending on the state of the system. When an agent must react to an event, it can decide to act immediately using available information, or it may prefer to wait in order to obtain more information. Indeed, the cost of un repaired

\[\text{No model of the system’s correct behaviour is available. The system can only use faults model, a priori known or automatically learned from the system observation.}\]
faults on the system increases with time. Consequently, it may be advantage-
ous to repair immediately. Nevertheless, the information – local or coming
from other agents – obtained during the wait can increase the reliability of
diagnosis, thus avoiding triggering erroneous repairs.
As discussed below, although different works address the problem of decision
making under uncertainty, they do not apply in our context.
2. (Diagnosis inference) In a multiple-fault context, several diagnoses may,
for one agent and at a given time, explain a given sequence of events. Thus,
for an agent, to make a diagnosis and trigger the associated repair require
being able to autonomously reduce the set of potential explanations. Works
dealing with the diagnosis inference are mainly focused on the single-fault
context [CPTMV07], and the literature addressing the multiple-fault case
[DMP07,DKW87] does not integrate the repair phase. The combination of
the diagnosis and repair capabilities is thus unexploited to reduce the un-
certainty associated with certain situations.
3. (System consistency) The supervision system works in a context of asyn-
chronous and unbounded communications. In this context, the theorem of
Fisher-Lynch-Paterson [FLP85] states the impossibility to guarantee the
achieving of a consensus between different agents. Thus, in case of com-
munication malfunction, opposing decisions taken unilaterally by agents can
cause disturbance in the supervised system. A mechanism for the resynchro-
nization of the system state as it is perceived by agents must be considered.

This article addresses the first point of this architecture, that is the agent’s
decision model.

The problem of online decision-making under uncertainty is the central point
of the work by Horvitz [HR91], Hansen et Zilberstein [HZ01] on the control of
anytime algorithms. The first distinction between these works and ours is that
their proposals require knowing a priori the optimal solution (or an estimation)
and to be able to dynamically determine the distance between the current so-
lution and the optimal one. In our work, the distance notion is not applicable.
Indeed, a diagnosis is right or wrong, and its “value” is only known a posteriori.
The second point of divergence is that we try to select a candidate (a diagnosis
or a repair) among a set of potential solutions. Indeed, the number of poten-
tial diagnoses is exponential in the number of known faults [DKW87], and the
evolution of the number of candidate diagnoses over time relies on the time and
on the state of the environment. The complexity of the task is therefore increased.

The agent decision model that we propose in this article allows to select
a diagnosis among a set of potential explanations by reaching a compromise
between the risk of error and the cost of inaction. This mechanism allows the
supervision system to dynamically adapt its behaviour to the communication
state. This paper is structured as follows. We first introduce our fault-and-repair
model and the various assumptions we have made in section 2. We then describe
in detail the proposed decision model in section 3. We present and discuss the
experimental evaluation of our decision model in section 4. Finally, we discuss current limitations of our approach and some perspectives of this work in section 5.

2 A Multi-Agent Architecture for the Supervision of Distributed Systems

The supervision process is distributed among several autonomous agents having each a local view of the system to be supervised, and endowed with diagnosis and repair capabilities. The supervised system is partitioned into regions, each one is supervised by one agent. As illustrated in Fig. 1, the supervision agents \( A_i \) exchange information in order to establish a diagnosis and a repair consistent with the observations \( O_j \) they get from the various units of the supervised system \( U_k \).

![Fig. 1. Simplified representation of the digitizing chain of the Dem@tFactory project. The links between the units represent the standard workflow. The dashed arrows represent the fact that some documents may be reprocessed if the quality is not sufficient. The links between the units and the agents represent the communication links used by the units in order to transmit alarms logs. The remaining links represent the communications between the supervision agents.](image)

2.1 Assumptions

We consider that communications are asynchronous and that there is no upper bound on transmission delay. We assume that the messages exchanged between supervised units may be lost or corrupted, and that some units are not supervised (e.g. unit \( U_2 \) on Fig. 1). This assumption is based on the fact that a complex industrial process commonly involves different partners\(^3\) that do not share their supervision information. Moreover, we assume that the observations and the

\(^3\) subcontractors in the case of the DematFactory project.
messages between agents can be lost but not corrupted. The agents are supposed to be reliable (no Byzantine behaviour [TVS02a]). Finally, we consider that the simultaneous occurrence of different faults does not result in phenomena of masking observables and we restrict ourselves to repairs that can be performed an infinite number of times. This last hypothesis is a restriction that permits to simplify the model in order to obtain clearer explanations while keeping the complexity degree unchanged.

2.2 Fault model and repair plan

Let \( F \) be the set of known faults of a system \( S \) and \( R \) be the set of existing repair plans. The signature of a fault \( f \) is a sequence of observable events generated by the occurrence of \( f \). The set of signatures of a given fault \( f \) is \( \text{Sig}(f) \).

Each fault is associated to a \( t \)-temporised Petri nets that represent its temporal dynamic (Figure 2). Each fault is supposed to be repairable, it is to say that there exists a least one partially ordered sequence of atomic repairs \( r_k \) that repairs it (a repair plan).

![Figure 2](image)

**Fig. 2.** Let \( f \) be a fault that possesses 2 signatures. \( \text{Sig}(f) = \{o_1, o_2, o_3\}^{[t_{o_1}, t_{o_1} + 5'']} \). The \( o_i \) are the events observed on the supervised system. The \( t_{o_i} \) indicate the temporal constraints. Thus, \( [t_{o_1}, t_{o_1} + 5''] \) constrain the sequence of observations \( [o_2, o_3] \) to appear under the 5 seconds that follow the occurrence of \( o_1 \) for \( f \) to be recognized.

The supervised system is partitioned into regions \( r_g_j \). Each supervision agent is associated with one unique region and knows the models of the faults that may occur in the region it oversees. However, a fault can cover several regions. In that case, an agent only knows the part of the model that concerns its region. Its model is completed with the names of the agents responsible for the others regions. Thus, from the location of the observations \( o_i \) of a fault, we can build the signatures associated to the agents:

\[
\text{Sig}(f) = \{o_1^{r_{gb}}, o_2^{r_{gc}}, o_3^{r_{gc}}, o_4^{r_{gc}}\} \Rightarrow \begin{cases} 
\text{Sig}_{A^{rg}(f)} = o_1 o_2 A^{rg} \\
\text{Sig}_{A^{rg}(f)} = A^{rg} o_2 o_4 
\end{cases}
\]

Beyond the obtaining of the models of faults, the difficulty of defining a global precedence relation between events that occur within the supervised system remains. Indeed, there is no common clock to the different regions. It is therefore necessary to add in each supervision agent a stamping mechanism allowing to
recreate this order relation. We will not detail here the concept of distributed clock \cite{TVS02}. We consider in the remainder of this paper that the agents are able to recreate this partial-order relation.

2.3 Diagnosis and multiple faults

During the period of time \([t - \Delta_t, t]\), agent \(A_i\) collects a sequence of observations \(seqObs_{A_i}(t, \Delta_t)\) generated by the occurrence of faults on the system. However agent \(A_i\) does not know which faults have occurred. It thus analyses \(seqObs\) in order to determine the set of all faults \(fp_{A_i}(t, \Delta_t)\) whose signatures partially or totally match elements of \(seqObs\). A diagnosis \(dg\) is a set of faults that can explain \(seqObs\). \(Dg\) is the set of all possible diagnoses of \(seqObs\).

2.4 Fault cost and repair cost

Finally, each fault \(f\) (respectively each repair plan \(rp(f)\)) is associated to a cost of dysfunction which depends of the fault duration \(Ct_{dysf}(f, t)\) (resp a cost of execution \(Ct_{Ex}(rp(f))\)). The cost of a diagnosis \(dg\) for the supervision system is the result of the aggregation of the respective costs of the faults that compose it. In the general case:

\[
Ct_{dysf}(dg, t) = \text{Aggreg}_{f_j \in dg}(Ct_{dysf}(f_j, t))
\]  

Similarly, the execution cost of a repair plan \(rp\) associated to a given diagnosis depends on the aggregation of the respective costs of the repairs that compose it. Thus, in the case the repair plan depends directly on the faults:

\[
Ct_{Ex}(rp(dg)) = \text{Aggreg}_{f_j \in dg}(Ct_{Ex}(rp(f_j)))
\]

3 Agent Decision Model

We consider highly dynamic systems. Consequently, information available to an agent at a given time can be insufficient to determine with certainty which action to select. An agent must be able to assess the desire of immediately triggering the plan made under uncertainty, compared with waiting and communicating with other supervisor agents in order to increase the reliability of its decision.

A supervision agent has to determine the optimal decision \((D_{opt})\) choosing between an immediate repair action \((Dimm_{opt})\) and a repair action considered in \(k\) time steps \((Delay_{opt})\). This waiting time can yield information that reduces uncertainty and thereby improve decision-making. The counterpart is that the elapsed time may have a significant negative impact on the system. The expected potential gain in terms of accuracy must be balanced with the risks taken.

Let \(Ct(x)\) the cost of an action \(x\) and \(Ct_{wait}(k)\) the cost related to the extra time \(k\) before selecting a repair plan. The decision-making process of each supervision agent works as follows:
1. Observation gathering
2. Computation of the different sets of faults that can explain the current observations : $Dg$ (set of diagnosis)
3. Sorting of the different diagnoses appearing in $Dg$ based on available information and on the constraints we chose to focus on (Most Probable Explanation, Law of parsimony, Worst case,...)
4. Determination of the immediate repair $Dimm_{opt}$ and computation of its estimated cost $Ct(Dimm_{opt})$
5. Computation of the waiting cost $Ct_{wait}(k)$, determination of the delayed repair $Ddelay_{opt}$ and of the associated cost $Ct(Ddelay_{opt})$.
6. Choice between the immediate repair $Dimm_{opt}$ and the delayed repair one $Ddelay_{opt}$

We will detail in the following sections the steps 4 and 5 relative to the determination of the immediate and delayed repair and of their respective costs.

### 3.1 Determination of the immediate repair $Dimm_{opt}$

The knowledge of the different signatures of faults allows us to establish a list of potential diagnoses $Dg$. We sort these explanations according to available information and to the constraints we chose to focus on (e.g the most probable explanation). After this step, the first element of $Dg$ is the diagnosis considered as the most relevant at the current time. It is then necessary to estimate its cost.

The cost of the immediate repair $Ct(Drep_{opt})$ must take into account the execution cost of the repair plan associated to the diagnosis retained ($Ct_{Ex}$, equation 2 page 6), as well as a cost representative of the potential error relative to this decision, $Ct_{Err}$. Indeed, if the only cost considered is the one of the execution of the repair plan, the final decision (step 6) will always favour an immediate action compared with a delayed one due to the additional waiting cost of the delayed action

$$Ct(rp(dg_1)) = Ct_{Ex}(rp(dg_1)) + Ct_{Err}(dg_1, Dg\{dg_1\})$$

The computation of the error cost $Ct_{Err}$ relies on the fact that we assume that the good diagnosis – and so the good repair – belongs to the sorted list $Dg$ of the potential diagnoses. Thus, in case of misdiagnosis when selecting the first diagnosis $dg_1$ of $Dg$, the system will lose a time equal to the execution time of the first repair plan ($Ct_{ExecTime}(rp(dg_1))$) which will be supplemented by the execution cost of the newly chosen repair plan ($Ct_{Ex}(rp(dg_2))$) associated to the 2nd diagnosis of $Dg$. As this second choice may also turn out to be an error, we define $Ct_{Err}$ recursively on $Dg$. Thus:

$$Ct_{Err}(dg_1, []) = 0 // Dg is empty, the diagnosis is correct.$$  

$$Ct_{Err}(dg_1, dg_2 :: Dg) = P(dg_1, dg_2 :: Dg) \times$$

$$\left[ Ct_{ExecTime}(rp(dg_1)) + Ct_{Ex}(rp(dg_2)) + Ct_{Err}(dg_2, Dg) \right]$$

$$Ct_{Err}(dg_2, Dg)$$

(4)
with $P(dg_1, dg_2 :: Dg)$, the probability – considering the other possible diagnoses – of choosing $dg_1$ as the final diagnosis to be an error.\footnote{Without additional information, $P(dg_1, dg_2 :: Dg) = |dg_2 :: Dg|/|dg_1 :: Dg|$}

### 3.2 Determination of the delayed repair $D_{delay_{opt}}$

A time $t$, an agent knows the set of the faults that may be occurring in the region it supervises $fp_{A_i}(t, \Delta_t)$. The different faults models are represented using t-temporised Petri-nets (Figure 2 page 5). The agent is thus able to predict, for each fault of $fp_{A_i}(t, \Delta_t)$, the set of observables that can be expected to appear during the time interval $[t, t+k]$, with $k$ an $a \ priory$ fixed parameter.

Note that the agent uses the current transmission duration (computed over the interval $[t-\Delta_t, t]$) to determine the set of potential observations.

From this information, the agent builds the tree representing the set of all possibles futures working towards the current time plus $k$ units of time, $Arb_{possibles}^A(k)$. Each node of the tree is associated with a set of observations and represent one possible future (Figure 3 below). The agent then computes, for each node of the tree, the set of diagnoses that explain this future ($Dg'$).

![Fig. 3. Illustrative example of a tree of the possibles futures.](image)

The agent can then compute, for each possible future, the immediate decision considered as optimal. At time $t$, the determination of the delayed decision with horizon $k$ ($D_{delay_{opt}}$) amounts to choosing between the various possibles situations. This choice is realised by sorting the first elements of each $Dg'$ of the tree of the possibles futures with each other using the same criteria than those used to sort $Dg$ in order to identify the immediate repair in the sub-section 3.1.

Note that in a multiple-fault context – assuming a total overlapping of the observations that compose the different faults of the supervised system and a finite horizon – the size of the tree of possibilities to explore is $O(2^{|o_i|})$. Associated with the limited resources of the agents, this complexity does not allow to completely explore the search space and requires adapting the windows’ size...
Once the delayed decision is identified, its cost $C_t(D_{delayopt})$ is established using Equation (3). We then have to add to this cost the waiting cost $C_t\_wait$. This waiting cost represents the consequences of the faults on the supervised system during the time where no action was triggered. The computation of the waiting cost depends on the respective costs of the malfunctions associated to the remaining diagnoses and of the elapsed time.

$$C_t\_wait(k) = Aggreg_{dg_i \in D_g}(C_t\_dysf(dg_i, k))$$ (5)

### 3.3 A Robust Approach

In cases where a significant amount of data (or a prior model) is available, probabilistic models are one possible approach to the forecasting of supervised systems’s behaviour [DGH92]. The major drawback of such models is their propensity to retain only the most probable hypothesis at the expense of the other possible hypotheses to get the optimal decision. Indeed, while such behaviour may be satisfactory from a macroscopic point of view, the consequences of highly suboptimal decisions – even if they are not very likely – may be unacceptable.

In an industrial context such as the one of the Dem@tFactory project, it seems so essential to consider the utility of a decision according to the cost of the occurrence of a given set of faults rather than its occurrence probability. This reasoning led us to favor a robust criterion [KY97] for the determination of the different aggregation’s functions $Aggreg$ used so far.

The decisions taken by the agents will therefore rely on the worst case hypothesis. Thus, the waiting cost $C_t\_wait$, the execution cost $C_t\_Ex$ and the cost of the consequences of an inefficient repair on the system $C_t\_ExecTime$ use the max function. The criterion used to sort the $D_g$ and $D_g'$ is the cost of the consequences of the diagnoses in the worst case. Finally, $D_{repopt}$, $D_{delayopt}$ and $D_{opt}$ minimize the cost of repair in case of misdiagnosis.

Algorithm 1 below determines all the candidate diagnoses and then establishes the various intermediate decisions before choosing the one that achieves the best balance between risk and cost. It is executed at each time-step and by each agent when faults occur. The value $k$ represents an upper bound as an agents’ decision is updated each time an observation is received.

### 4 Experimental Evaluation

To evaluate our approach we have developed a simulator for distributed systems. Based on the JADE multi-agent platform [CGK+05], our environment allows us to model both physical units and communications links, and to simulate the occurrence of failures in it. For a given simulation, a list of faults is associated
Algorithm 1 Anytime Decision

Require: seqObs, F, R; Ct(), k
1: while true do
2: Update fp, Dg
3: Sort Dg
4: Rp_t = \{r \in R|\exists dg \in Dg, r \in Rp(dg)\}
5: Ct_{wait}(k) = \max_{dg_j \in Dg} (Ct_{dysf}(dg_j, k))
6: Dimm_{opt} = \min_{rp \in Rp, dg_i \in Dg} \max_{dg_j \in Dg} (Ct_{Ex}(rp(dg_i)) + Ct_{Err}(Dg))
7: D_{del_{opt}} = \min_{r \in R, dg_i \in Dg} \max_{dg_j \in Dg} \max_{dg_j \in Dg_{+k}} (Ct_{Ex}(rp(dg_i)) + Ct_{Err}(Dg_{+k}))
8: D_{opt} = \min(Ct(Dim_{opt}), Ct_{wait}(k) + Ct(D_{del_{opt}}))
9: Apply(D_{opt})
10: end while

with each site and each communication link (A communication link can increase its transmission time, a unit may stop working properly,...). Each fault is associated with one or more trigger conditions: a date and/or the occurrence of another failure. This last point allows us to simulate cascading faults. Our supervision system is deployed on this simulator. When running the simulation, some faults trigger the sending of an alarm message to the agent responsible for the site where they appear. These messages are the observations from which the agents will try to determine the appropriate behaviour.

The upper bound k is set to 15 units of time. Moreover, we assume in these experiments that the respective costs of the faults that compose a diagnosis are additive. Finally, in order to have benchmarks for the evaluation of the principles underlying our architecture (RDS2), we also implemented a centralized supervision systems (SC) where all observables are transmitted to a single supervision agent.

4.1 Experimental protocol

In this experiment, we want to study the behaviour of the different supervision systems when varying the transmission cost (assumed homogeneous) of the information for a given topology (Figure 1 page 4). We fixed a priori the locations and responsibilities (the regions) of the supervision agents according to the geographical location of the units that compose the supervised process.

We performed 5 simulations for each value of the transmission time (between 0 to 30 units of time). We use a statistical model to trigger the faults during the simulations. We used the dataset brought by our industrial partners (8 GB of data corresponding to 48 hours of logs) and extracted nineteen different faults that we have classified in two categories: primary faults and secondary faults. The primary faults may occur in the absence of prior faults in the supervised system. The secondary faults occur only on the condition that one or more faults
are already present in the system. From information gathered from our partners, we were able to estimate the costs of the faults over time (constant, logarithmic,...) and the associated repair costs. We determined that the probability of occurrence of \( n \) primary faults per unit of time follows a Poisson distribution with parameter \( \lambda = 0.043 \). The secondary faults are automatically triggered when their trigger conditions are met.

The performance evaluation is based on three criteria: (1) The average response time to a malfunction. (2) The average number of supervision messages exchanged. (3) The average total cost of repairs made during the experiment.

4.2 Results

Figures [4] and [5] present the evolution of the behaviour of supervision systems RDS2 and SC for the two first criteria. The vertical bar at \( t=15 \)mins is the horizon considered by the agents of RDS2 for the computation of the delayed decision.

**Fig. 4.** Evolution of the average response time of RDS2 and SC when varying the transmission delay.

**Fig. 5.** Evolution of the average number of supervision messages of RDS2 and SC when varying the transmission delay.

Figure 4 shows the average response time to a malfunction of the supervision systems SC and RDS2 as a function of the communication state. The slowing of the communication firstly has a more important impact on RDS2 than on the centralised supervision. The number of supervision messages needed by RDS2 to determine the best action to perform is the main reason. However, the response time of RDS2 progressively stabilize around 15mins. On the contrary, the response time of SC increases over time and becomes higher than the one of RDS2 when the transmission delay goes past 6 mins. This can be explained by the fact that the agents of RDS2 can decide to act without waiting for the
reception of all the messages that come from the units of the supervised system. We can observe another increase of the average response time of $RDS_2$ when the transmission delay is close to 15 mins. In that case, the response time stabilizes around 21 mins. This second increase is due to the parameter $k$ of our algorithm, a priori fixed to 15 mins. This parameter defines the agent’s horizon for the computation of the delayed decision. When the transmission time becomes greater or equal to $k$, an agent no longer has interest to wait or to try to exchange information with other agents of $RDS_2$; so it decides to act at the risk of making a mistake. This behaviour is clearly highlighted by the graphic representing the average number of supervision messages exchanged.

Indeed, as shown in figure 5, the number of messages exchanged by the agents of $RDS_2$ drop with the increase of the transmission delay. We can also observe a sudden drop of this number when the transmission delay becomes greater to 15 mins, confirming the local decisions-making of the agents. We may note that the number of supervision messages of $SC$ progressively grows with the increase of the transmission delay. This growth is explained by the diagnoses errors made by the centralized supervision system due to the delays when receiving messages from the units that compose the supervised system.

The third graphic (Figure 6) presents the evolution of the average cost of the repairs as a function of the communication state. We observe a progressive increase of the repair costs for $SC$ as well as for $RDS_2$. In this latter case, even if the costs are greater than those of $SC$, they seems to be stabilizing after than the 15mins bar is crossed.

Based on these experiments, we can conclude that the proposed decision model allows the agents to dynamically and effectively adapt their behaviours to the current state of the supervised system. Considering the reactivity of $RDS_2$ and the limited repair extra-cost it generates in case of communication failures,
the communication extra-cost for low transmission delays can be considered as an acceptable consequences compared with a total-absence of supervision.

5 Conclusion and Future work

In this paper, we presented a multi-agent approach and proposed an agent decision model in order to guarantee an anytime supervision of distributed systems that face unreliable or costly communication links. Our decision model allows the agents to find a balance between an immediate repair under uncertainty and a delayed action, more reliable but which delay can engender important consequences on the supervised system. Each agent is thus able to dynamically adapt its diagnosis-and-repair process to the state of the supervised system. Our results, obtained using a real dataset, show that our proposal is able to perform an efficient adaptive supervision for systems where a short response time prevails over a limited extra-cost for the repair.

Future work will first deal with performing a finer grained evaluation of our approach. Specifically, the results obtained with real data presented in this paper combined with the evaluation of our approach on various environments topology \cite{HCEFS11} will lead us to study the impact of the parameter $k$, which defines the agent’s prediction horizon, and of heterogeneous communication costs on the performances on our distributed supervised system. More in detail, while in this work we assumed homogeneous communication costs, these are generally heterogeneous in real environments. We expect that this heterogeneity will allow the agents to exploit efficiently the inherent distribution of the multi-agent system and thus improve the supervision performances.

Regarding the limitations of our proposal, the theoretical complexity of the tree of possibilities presented in section 3.2 is an important point. In the dataset brought by our industrial partners, the overlapping rate of the observations that compose the known faults is relatively low (less than 13%). The practical complexity of the tree is thus reduced. However, a dataset with a more important overlapping rate can exist, and will lead to the exploration of a much larger search space in order to make a diagnosis. This will consequently increase the risk of misdiagnosis by the agents.

As presented in the introduction, our architecture is composed of three components, and this paper only present the agent decision model. Dealing with the size of the diagnosis search space in case of a high overlapping rate is the aim of the two other components of our architecture, the diagnosis inference mechanism and the consistency mechanism. The diagnosis inference mechanism uses the fact that each agent can perform both diagnosis and repair process. It extends the work of \cite{CPTMV07} to the multiple-fault context and interleave the diagnosis and repair phases to reduce the size of the search space and obtain the minimal set of diagnoses. The consistency mechanism, deals with the necessity for the agents to reach a consensus on the decision to retain at the risk to
lead to serious inconsistencies and malfunctions in the system. We are currently implementing these components in the presented architecture in order to realise a practical analysis of their impact on the effectiveness of the supervision system.

Acknowledgements

The authors would like to thanks Sylvain Ductor and the anonymous reviewers of this paper for the quality of their comments and suggestions.

References


