Institut pour la Maîtrise des Risques
Sûreté de Fonctionnement - Management - Cindyniques

Réseaux de Pétri et études de survivabilité des systèmes de systèmes

Petri Nets and systems of systems survivability studies

LEBOISSELIER Thaïs
DGA
Bruz
thais.leboisselier@intradef.gouv.fr

Résumé — Cette communication présente une étude sur la survivabilité des systèmes de systèmes (SdS) dans des contextes où des agressions intentionnelles peuvent survenir. L'objectif principal est d'évaluer la capacité du système de systèmes à préserver sa mission et à maintenir la survie de ses composants. Pour cela, l'étude se concentre sur l'utilisation des réseaux de Pétri, un formalisme de modélisation qui permet de représenter graphiquement et mathématiquement le comportement dynamique des systèmes. En utilisant des simulations et des analyses temporelles, l'étude compare deux architectures de système de systèmes, centralisée et distribuée, face à trois événements redoutés liés à la communication. Les premiers résultats indiquent que l'architecture distribuée est plus robuste dans ces scénarios spécifiques, mais soulignent également la nécessité d'explorer davantage de paramètres et de considerations comme l'exploration d'architectures mixtes et en augmentant le nombre de systèmes examinés pour parvenir à des conclusions définitives. Enfin, l'étude identifie les limites des outils de recherche et envisage l'utilisation d'outils commerciaux pour des analyses plus poussées.

Mots-clefs — survivabilité, système de système, modélisation, militaire, réseau de Pétri.

Abstract — This communication presents a study on the survivability of system of systems (SoS) in contexts where intentional aggressions may occur. The main objective is to evaluate the capability of the system of systems to preserve its mission and maintain the survival of its components. To achieve this, the study focuses on the use of Petri nets, a modeling formalism that allows for graphical and mathematical representation of the dynamic behavior of systems. Using simulations and temporal analyses, the study compares two architectures of system of systems, centralized and distributed, facing three feared events related to communication. Initial results indicate that the distributed architecture is more robust in these specific scenarios, but also highlight the need to further explore parameters and considerations such as the exploration of mixed architectures and increasing the number of systems examined to reach definitive conclusions. Finally, the study identifies the limitations of research tools and considers the use of commercial tools for more in-depth analyses.

Keywords — survivability, system of system, modelisation, military, Petri net

I. INTRODUCTION

La survivabilité est l'étude de la capacité d'un dispositif opérationnel donné à assurer sa mission et à préserver la survie de ses constituants dans un contexte potentiellement marqué par des agressions intentionnelles perpétrées par des menaces.
L'un des axes retenus pour améliorer la connaissance dans le domaine de la survivabilité des systèmes de systèmes (SdS) est d'exploiter la modélisation de systèmes dynamiques, au cœur de la transition numérique de la Direction Générale de l'Armement (DGA) qui s'oriente vers plus de modélisation et de simulation au cœur de son activité d'équipement des forces armées. L'objectif est de limiter les risques d'agressions et leurs conséquences sur nos forces déployées et d'augmenter ainsi la résilience et la survie de notre armée.
Le projet s'inscrit dans le cadre des projets étudiants de l'école d'ingénieurs IMT Nantes Atlantique et de l'école d'ingénieurs ENSIBS, projets imaginés et pilotés par la DGA. Il prépare le futur, dans le cadre de système de systèmes (SdS) collaboratifs ; alors que de nouveaux types d'agressions émergent avec l'essor des nouvelles technologies : cyber attaque couplée avec une attaque physique ; intelligence artificielle (IA) accélérant la détection et l'interprétation de menaces potentielles…
La survivabilité des SdS est un sujet innovant et l'outillage des études est encore limité dans ce domaine. Les réseaux de Pétri semblent être pertinents pour étudier ce domaine d'activité. L'ambition du projet d'étude permet, sur un exemple concret, de

vérifier l'utilité de ce type d'outil pour le domaine de la survivabilité, et de tester différents outils pratiques, libres ou du commerce, sur la réalisation et l'analyse des réseaux de Pétri.

Tout d'abord, un état de l'art sur les réseaux de Pétri sera présenté. Ensuite, la méthodologie utilisée et les résultats obtenus seront exposés. Pour finir, les perspectives et une conclusion ouvriront le sujet sur une suite possible.

II. ETAT DE L'ART

A. Historique des réseaux de Pétri

Les réseaux de Pétri (abrégé RdP) sont un formalisme de modélisation inventé en 1962 par Carl Adam Pétri. Leur but premier est de proposer une méthode de représentation des systèmes concurrents et des effets de leur comportement dynamique des événements discrets. Ils sont considérés comme les ancêtres des Grafcet (1977) ou encore des diagrammes d'activités UML (Unified Modeling Language) (1997), qui en sont des dérivés simplifiés.

Il s'agit d'un outil de modélisation puissant et plutôt répandu dans le domaine de la recherche et du développement pour des problèmes d'évaluation de systèmes complexes. Les réseaux de Pétri permettent essentiellement la modélisation sous forme de graphe de systèmes dans de nombreux domaines d'application (informatique, télécommunication, biologie...). Dans l'industrie, les réseaux de Pétri sont de plus en plus utilisés, notamment pour les études de disponibilité de production [1].

Les réseaux de Pétri sont donc particulièrement adaptés lorsqu'il s'agit de modéliser des systèmes nécessitant la prise en compte de la concurrence, la synchronisation, le partage de ressources et l'exclusion mutuelle. Un de leurs principaux avantages est la double représentation qu'ils offrent, à la fois graphique et mathématique.
Étant également très extensibles, de nombreuses variations du modèle réseaux de Pétri sont apparues au cours des années. Certaines limitations, comme l'impossibilité de différencier les jetons, peuvent exister dans la version originelle des réseaux de Pétri. Cela a conduit notamment à la création de réseaux de Pétri colorés.

1) Représentation classique

Un réseau de Pétri est un modèle mathématique permettant la représentation d'une variété de types de systèmes sous forme de graphe orienté. Il est composé de places, de transitions, de jetons et d'arcs et possède un état initial.

Les places (P) sont représentées visuellement par des cercles et permettent généralement d'illustrer des états. Chaque place contient un certain nombre de jetons modélisés par des points noirs qui représentent les ressources détenues. La disposition des jetons dans les places constitue le marquage du graphe ; leur disposition initiale étant quant à elle appelée le marquage initial. Enfin, le nombre de jetons pouvant être contenus dans une place peut éventuellement être limité par une valeur entière appelée la capacité de la place (variable k sur l'image ci-contre). Les transitions (T), représentées sous forme de rectangles, permettent de décrire des événements qui modifient l'état du système en distribuant les jetons. Une transition représente bien souvent une action qui a été déclenchée par un acteur interne ou externe et qui permet d'envoyer des jetons d'une place à une autre, c'est-à-dire faire basculer le système d'un état à un autre.

Les arcs sont représentés par des flèches et symbolisent de manière générale les entrées et sorties de jetons. On peut également dire qu'ils indiquent les pré-conditions et post-conditions requises par une transition pour que celle-ci soit franchissable. En effet, dans ce type de graphes, les jetons peuvent se déplacer d'une place à l'autre par le biais des transitions et des arcs qui y sont liés. Une transition est dite sensibilisée (ou activée) lorsque des jetons peuvent la franchir, autrement dit, lorsque les pré-conditions sont remplies. (Fig. 1)

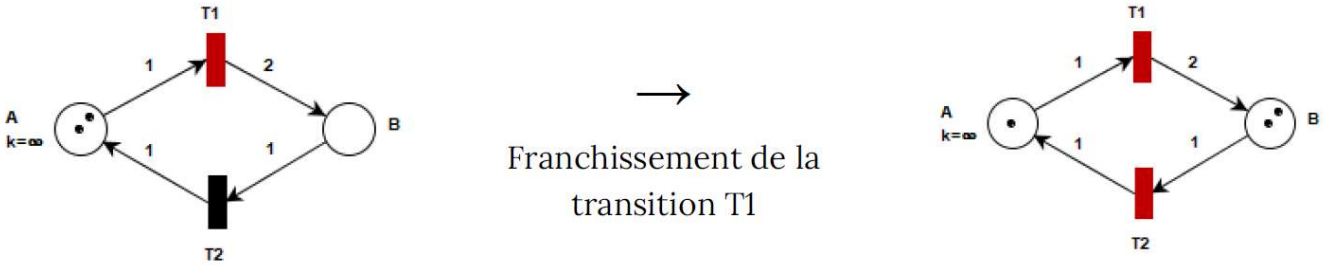

Fig. 1. Exemple d'un petit réseau de Pétri avec franchissement de la transition T1

Ces arcs peuvent donc lier une place à une transition (on dit alors que des jetons sont consommés) ou une transition à une place (les jetons sont produits) et sont associés à une valeur entière indiquant le nombre exact de jetons concernés : la pondération de l'arc. Il ne peut pas exister d'arcs entre deux places, ni entre deux transitions, faisant du réseau de Pétri un graphe biparti.

Il existe également un deuxième type d'arc appelé arc inhibiteur qui permet d'empêcher le franchissement d'une transition tant que des jetons sont présents dans une place spécifique.

2) Classification des réseaux de Pétri

Il existe de nombreux modèles de réseaux de Pétri permettant d'étendre ses propriétés de base. La présentation ci-dessous décrit les modèles principaux, à savoir SPN (Stochastic Petri Net), GPN (Generalized Petri Net) et CPN (Colored Petri Net). Il est cependant à noter que les propriétés énoncées ci-dessous peuvent être combinées entre elles pour aboutir à des modèles hybrides (dont le modèle GSPN pour Generalized Stochastic Pétri Net, qui est un des plus populaires).

a) Colored Petri Net (CPN)

Dans certaines situations, il est nécessaire de dupliquer certaines parties d'un réseau de Pétri afin d'illustrer un cas de figure particulier. Ainsi, pour représenter des systèmes plus complexes tout en évitant l'ajout de places et de transitions redondantes, l'utilisation de jetons colorés peut se révéler utile.
Dans les réseaux de Pétri colorés, les couleurs des jetons permettent de les catégoriser selon des critères relativement libres, définis par le concepteur du réseau. En regroupant les jetons par type d'usage, il est alors possible d'imposer des conditions plus précises pour le franchissement des transitions. Dans ce type de réseau, les arcs disposent désormais d'un poids pour chaque couleur de jetons. De plus, si la pondération d'un arc est nulle pour une certaine couleur, le passage de n'importe quel jeton de cette même couleur sera totalement interdit.
Les outils qui permettent la manipulation de réseaux colorés (PIPE, GreatSPN,…) proposent généralement des outils permettant de développer ces réseaux de Pétri sous la forme de réseaux non colorés. Les réseaux de Pétri générés de cette manière sont souvent très complexes et répétitifs dans leur structure, ce qui démontre bien l'intérêt des jetons colorés.

b) Generalized Petri Net (GPN)

Un réseau de Pétri généralisé désigne les réseaux pour lesquels les pondérations des arcs peuvent être variables, et donc différentes de 1. Ajouter une pondération spécifique aux arcs de part et d'autre d'une transition permet ainsi de préciser le nombre de jetons consommés et produits lors de son franchissement.

c) Timed Petri Net (TPN)

Dans un réseau de Pétri classique, les transitions sont toutes immédiates, autrement dit, associées à un délai de déclenchement nul. Un réseau de Pétri temporisé supporte l'ajout d'un nouveau type de transition permettant de spécifier les délais précédant et suivant les transitions, et plus particulièrement :

- Le délai requis entre l'activation d'une transition et le début de son franchissement.
- Le délai de franchissement de la transition.

d) Stochastic Petri Net (SPN)

En sciences, l'adjectif "stochastique" est utilisé comme un synonyme d'"aléatoire". Ce terme s'oppose à la définition de "déterministe", qui dans le cadre de la conception de systèmes, désigne le fait qu'il suffise de connaître les états précédents d'un système pour déterminer son état actuel et les suivants.
Dans un réseau de Pétri stochastique, les transitions sont franchies après un certain délai, défini selon une loi probabiliste (comme la loi uniforme ou exponentielle par exemple). Utiliser ce type de transition permet donc de prendre en compte les incertitudes liées à la temporisation, et de rendre la modélisation plus conforme avec la réalité.

B. Analyse de performance

La dissertation de Mark WOODARD ([2], p. 41), publiée en 2017, indique que par le passé, de nombreuses études ont proposé d'utiliser le temps de récupération d'un système (time-to-recovery ou TTR) comme métrique de mesure de survivabilité d'un système cyber-physique. Les résultats obtenus au cours de ces études ne permettaient pas cependant d'analyser le comportement du système de façon suffisamment précise, le TTR n'étant qu'une somme de la durée de panne et de la durée de réparation du système.
Face à ce constat, deux approches complémentaires sont présentées à travers les différentes suggestions dans la littérature. La première s'appuie sur un calcul de probabilités en fonction de l'état actuel du SdS et du temps. La seconde se repose sur une méthode de vérification de conditions à partir d'arbres décisionnels.

1) Approche stochastique et probabiliste

Pour modéliser la survivabilité, il est question d'utiliser les réseaux de Pétri comme un moyen de représentation des transitions entre les niveaux de condition opérationnelle d'un système (autrement dit, les passages entre les états dégradés, défaillant, détruit ou fonctionnel).

Différents algorithmes d'évaluation stochastiques peuvent alors être applicables. Ainsi, la méthode de Monte Carlo, qui consiste à répéter un tirage aléatoire un grand nombre de fois dans les mêmes conditions et de calculer ensuite la moyenne des mesures relevées pour obtenir une approximation d'un résultat. Dans le contexte des réseaux de Pétri, la méthode de Monte Carlo permet d'estimer les probabilités de présence de jetons dans les différentes places. Ainsi, de nombreux outils existants incluent un simulateur qui permet aux jetons de franchir automatiquement les transitions si celles-ci sont sensibilisées (et même de décider aléatoirement si plusieurs choix sont possibles) et de déclencher des séquences de tirages (ou en anglais "firings"). Par exemple, si dans notre système, une place correspond à un état défaillant, nous pourrons alors aisément prédire la probabilité du risque de panne.
Technologie de TotalEnergies depuis les années 1980, GRIF (Graphiques Interactifs pour la Fiabilité) comprend un module Petri de simulation permettant de modéliser le comportement de systèmes dynamiques complexes par réseaux de Petri stochastiques à prédicats et assertions. Ce module s'appuie sur MOCA-RP (pour MOnte-CArlo – Réseaux de Petri) : un moteur de calculs rapide basé, comme son nom l'indique, sur la simulation de Monte-Carlo et qui repousse les limites de la modélisation. La flexibilité du module permet d'obtenir à la fois les grandeurs usuelles de la sureté de fonctionnement (disponibilité, fiabilité, …) et les informations relatives aux systèmes de production (quantités produites, nombre de ressources utilisées, …). Cet outil ne permet pas la création de réseau de Pétri coloré, mais présente un module d'analyse des résultats performant.

2) Approche par logique temporelle et arborescente

Certains outils développés par des chercheurs, comme PIPE, proposent des outils d'évaluation des modèles basés sur des arbres de requête de performance (Query Performance Trees) ([3], p. 5). PIPE permet notamment la spécification de critères mathématiques organisés sous la forme d'un arbre décisionnel (PT) et affiche un aperçu de celui-ci en langage naturel. Cet outil peut permettre de vérifier automatiquement un certain nombre de conditions afin de s'assurer de la validité du modèle.
L'outil de vérification de modèle proposé par PIPE reprend les principes de la logique arborescente, qui consiste, dans le cas de l'analyse de graphes, à imbriquer des expressions de vérifications de chemins ou de temps. L'une des généralisations les plus populaires en termes de vérification formelle de graphes est la formalisation CTL (Computation Tree Logic). Elle s'inspire de la logique linéaire temporelle (LTL) initialement conçue en 1977 pour la vérification de programmes informatiques.
Dans le contexte de la vérification de réseaux de Pétri, LTL peut permettre simplement de compter le nombre de jetons présents dans une place ou encore vérifier que le graphe est deadlock-free. Les formules sont exprimées en logique propositionnelle booléenne tandis que la vérification de la temporalité peut être effectuée à l'aide de 4 opérateurs unaires additionnels : G ("globally"), F ("finally"), R ("release") et W ("weak until").
CTL ajoute au LTL la possibilité de vérifier l'ensemble des chemins d'un graphe à l'aide des opérateurs A (All) et E (Exist), équivalents aux prédicats logiques ∀ ("pour tout") et ∃ ("il existe"). Ces deux formalisations sont supportées par de nombreux outils d'analyse de réseaux de Pétri tels que Great SPN / StarMC Model Checker, TAPAAL ou encore TINA.
Quelques variantes ont également vu le jour, notamment pour essayer de cumuler les aspects logiques et temporels de LTL et CTL tout en conservant la possibilité d'analyser des réseaux non-déterministes (faisant donc appel au hasard et aux probabilités). C'est notamment le cas de la HASL (Hybrid Automata Stochastic Logic) utilisée par l'analyseur Cosmos depuis 2011.

III. METHODOLOGIE

L'idée générale du projet est de partir d'un outil connu de la sûreté de fonctionnement, les réseaux de Pétri, pour l'adapter au domaine de la survivabilité des systèmes de systèmes.
Les deux domaines sont proches, puisque la survivabilité des systèmes de systèmes regroupe la sûreté de fonctionnement (pannes et réparation), mais aussi les attaques cyber et les attaques conventionnelles (ex : tir de missile, de munitions).

A. Définition d'un système de systèmes

Un système de systèmes (SdS) est un ensemble de systèmes indépendants coopérant en vue d'une même finalité opérationnelle, et dans le but de rechercher un ensemble d'effets prédéterminés. La définition et la configuration de ce SdS sont évolutives, caractérisé par des comportements émergents (propriétés ou fonctionnalités spécifiques à l'assemblage) et constitués de systèmes distribués géographiquement possédant une indépendance opérationnelle et managériale : chacun des systèmes a sa propre raison d'être, et peut avoir un cycle de vie distinct. Les systèmes sont assemblés de façon organisée pour constituer une structure de niveau supérieur et leurs échanges, (d'informations) sont maîtrisés et coordonnés.
Exemples de SdS du domaine militaire: Groupe aéronaval, Sous groupement tactique interarmes, Système de surveillance aérienne.

B. Définition de la survivabilité d'un système de systèmes

La définition donnée au terme de survivabilité utilisée dans le domaine militaire [4] est la suivante :
La survivabilité d'un système de systèmes (SdS) est la capacité du SdS à accomplir sa mission (dans un état nominal ou dégradé) dans un contexte marqué par des événements indésirables, potentiels ou avérés, tout en préservant au mieux la survivabilité de ses systèmes constituants. Les événements indésirables peuvent être du type :

- agression intentionnelle,

- agression non intentionnelle provoquée par l'environnement naturel ou humain, un comportement émergent du SdS…,
- défaillance d'origine technique ou humaine au sein du SdS.

Cette capacité repose sur des choix techniques ou d'organisation du SdS et de ses systèmes constituants visant successivement ou simultanément à :

- limiter l'occurrence des événements indésirables (furtivité, moyens de dissuasion, dispositifs de prévention, sûreté de fonctionnement, maîtrise des comportements émergents…),
- limiter les effets de ces événements sur les hommes et sur les capacités du SdS essentielles à la poursuite de sa mission (moyens de protection, redondances d'équipement...),
- accroître le pouvoir de reconfiguration du SdS soumis aux effets de ces événements pour rétablir, complètement ou partiellement et dans un délai acceptable, les capacités essentielles à la poursuite de sa mission (restauration des systèmes, autorégulation, fonctionnement en modes dégradés…),
- procéder à un apprentissage permettant d'affronter les événements ultérieurs avec une efficacité accrue.

Cependant, la survivabilité d'un SdS n'est pas qualifiable de façon binaire étant donné qu'elle dépend de l'état de fonctionnement de l'ensemble des systèmes qui le composent. Il paraît donc plus juste de parler en niveaux de survivabilité.

En phase de conception, les équipes techniques ont plusieurs idées d'architectures de systèmes de systèmes qu'il faut comparer. La méthodologie des réseaux de Pétri est une méthode qui permettrait de comparer de manière qualitative et quantitative les différentes architectures, sur des critères dysfonctionnels de haut niveau.
Les arguments pour le choix des réseaux de Pétri sont qu'ils peuvent modéliser différents problématiques liées à la survivabilité d'un système de systèmes :

- Etat interne d'un système (bon état, défaillant, dégradé)
- Etat relatif d'un système parmi un système de systèmes AMI contre ENNEMI (non détecté, détecté, poursuivi par un radar ennemi, identifié par un ennemi, poursuivi par un missile ennemi, détruit)
- Communications entre systèmes (avec attaque ou interception cyber sur les messages, brouillage, problème de masquage dû au relief du terrain, problématique d'élongation entre deux chars trop distants pour pouvoir communiquer)
- Partage de ressources (exemple : réparateur, pièces de rechanges).

C. Cas d'étude

Le cas d'étude s'appuie sur un système de systèmes (SdS) simple dans un contexte militaire terrestre. Il s'agit plus particulièrement d'un système cyber-physique, c'est-à-dire d'un système composé d'entités physiques commandées et contrôlées par l'informatique. Il se compose de quatre types de systèmes interconnectés :

- Un drone aérien de surveillance
- Un char "Contrôle Commande"
- Un char "Canon"
- Un système de communication, à la fois embarqué dans les systèmes précédemment cités et déployé en extérieur.

L'étude à mener consiste à modéliser et évaluer le système de systèmes du point de vue de la communication ; l'objectif étant de garder le contrôle sur une zone et d'attaquer une cible ennemie, le tout en réduisant au maximum les risques pour permettre la survivabilité du SdS.

Pour limiter l'étude dans son périmètre, on considère dans la suite du document l'étude d'un SdS avec la configuration suivante : un drône aérien de surveillance, un char de contrôle-commande et trois chars canons. L'idée est de recueillir des premiers résultats sur ce petit SdS, mais de pouvoir étendre les résultats à un système de systèmes plus conséquent, avec 500 systèmes. Chaque char, que ce soit un char canon ou un char contrôle commande, embarque un système de communication composés de deux antennes, de deux radios, d'un serveur de communication et d'un système d'information (Fig. 2)

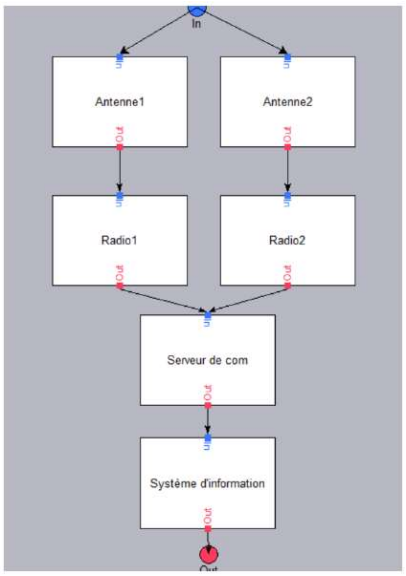

Fig. 2. Composants des chars

Chaque drone embarque un système de communication composés d'une antenne, d'un serveur de communication et d'un système d'information. (Fig. 3)

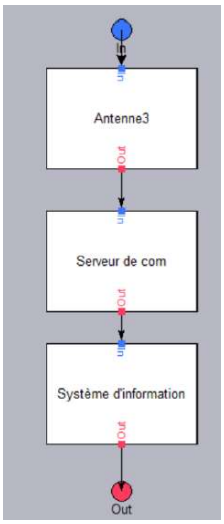

Fig. 3. Composants du drone

Les réseaux de Pétri permettent une représentation assez libre des systèmes. Selon le système étudié et le point de vue adopté, les places, les transitions et les jetons n'auront pas la même signification. L'association place-état et transition-action n'est pas systématiquement la plus adaptée, bien qu'il s'agisse d'un bon point de départ.
Dans l'étude de cas que nous étudions, les 3 évènements redoutés (ER) retenus sont :

- ER1 : Perte de communication

  La perte des communications peut être temporaire ou permanente, locale sur une partie seulement du SdS ou totale.

  Causes possibles : défaillance, masquage, élongation ou brouillage; erreur humaine

- ER 2 : Interception des communications

  Les communications sont interceptées par l'ennemi.

  Causes possibles : interception par l'ennemi d'un récepteur physique ; interception par l'ennemi par les ondes.

- ER3 : Intrusion dans les systèmes de communication à des fins de désinformation

  L'ennemi envoie de fausses informations : cela peut aller jusqu'au contrôle des entités autonomes.

  Causes possibles : intrusion par l'ennemi d'un récepteur physique ; intrusion par l'ennemi par les ondes.

La question posée par les architectes du système de systèmes est la suivante : « d'un point de vue de la survivabilité, quelle architecture est la meilleure entre les architectures suivantes : centralisée, distribuée, (Fig. 4) ou un mixte entre les deux précédentes architectures ? »

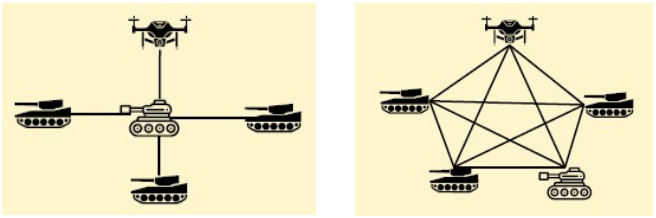

Fig. 4. AR1 : Architecture centralisée et AR2 Architecture distribuée

- AR1: Une architecture de type centralisée : la communication passe par un nœud central, ici le char contrôle-commande.
- AR2: Une architecture distribuée : la communication peut passer entre chaque système, par plusieurs chemins possibles, permettant ainsi la possibilité de s'affranchir de la perte d'un élément.

La question, simple à démontrer d'un point de vue de la disponibilité des réseaux, n'est pas triviale d'un point de vue de la survivabilité. Dans le domaine des réseaux informatiques, on rencontre traditionnellement deux types d'architectures : peer-to-peer et client-serveur.
Un réseau client-serveur est un type de réseau informatique dans lequel les données sont stockées et gérées par un serveur central, se rapprochant de l'architecture centralisée AR1. Les principaux avantages du réseau client-serveur sont généralement la sécurité qui sera renforcé pour le serveur central, et la scalabilité (il est possible de connecter des milliers d'utilisateurs sans affecter la vitesse ou la performance du réseau). Cependant, le principal inconvénient du réseau client-serveur est sa dépendance à un serveur central. Si le serveur tombe en panne, les ressources affectées peuvent s'arrêter et tous les utilisateurs connectés seront affectés.
Un réseau peer-to-peer ou P2P est un type de réseau informatique dans lequel chaque nœud du réseau (ordinateur, appareil mobile) fonctionne à la fois comme un serveur et un client, se rapprochant de l'architecture distribuée AR2. Cette typologie réseau offre plusieurs avantages, dont la connexion directe entre les différents utilisateurs et l'absence de serveur central. Les données sont stockées sur plusieurs serveurs, ce qui permet une redondance et une disponibilité optimale. Cependant, le principal inconvénient des réseaux peer-to-peer est qu'ils sont vulnérables aux attaques informatiques et au piratage de contenu, non étudiés en sûreté de fonctionnement. La survivabilité des SdS, incluant les attaques informatiques (avec l'ER2 interception des communications et l'ER3 intrusion dans les systèmes dans notre étude) devrait donc présenter des résultats différents.

D. Représentation du niveau de condition opérationnelle

Plusieurs groupes d'étudiants ont travaillé sur le même sujet. Les premiers groupes ont été confrontés à l'utilisation de logiciels libres, limités dans les calculs. La DGA avait orienté le sujet vers les réseaux de Pétri Colorés, prometteurs pour passer à l'échelle de 500 systèmes. En effet, l'inconvénient de ce type de modélisation est l'explosion de l'espace d'état : à mesure que le système grandit, l'espace d'états augmente exponentiellement, ce qui limite la lisibilité du réseau de Pétri et qui demande plus de capacité de calcul. Les réseaux colorés permettent de limiter le nombre de places sur ce type de configuration, et donc de palier à cette explosion d'états. Ces groupes n'ont pas eu le temps de conclure : ils ont dû passer du temps à coder pour pouvoir analyser leur réseau de Pétri. Etant donné que la facilité et la lisibilité des résultats de calculs étaient deux critères importants, les réseaux de Pétri colorés ont été abandonnés. Un autre groupe d'étudiants a pu essayer de modéliser ce sujet sur le logiciel GRIF, module Pétri, qui a l'avantage de pouvoir analyser les résultats facilement, mais qui ne présente pas la possibilité de faire des réseaux de Pétri colorés. Ce sont les résultats de ce dernier groupe qui sont présentés dans la suite du document.

E. Modélisation

La suite du projet a été réalisée avec un réseau de Pétri stochastique hiérarchique pour représenter le système de systèmes (SdS). Cette variante des réseaux de Pétri permet de répondre aux besoins spécifiques du SdS en utilisant différents réseaux de Pétri pour modéliser chaque système. L'utilisation d'un réseau de Pétri par système permet leur réutilisation dans des réseaux de Pétri de niveau supérieur, et ainsi composer des systèmes de systèmes plus simplement. L'utilisation des réseaux de Pétri stochastiques nous permet d'ajouter des fréquences et des probabilités sur les transitions, ce qui est important pour représenter de manière réaliste les événements qui peuvent se produire sur le SdS lors d'une mission. L'inconvénient de ce type de modélisation est l'explosion de l'espace d'états : à mesure que le système grandit, l'espace d'états augmente exponentiellement, ce qui limite la lisibilité du réseau de Pétri et qui demande plus de capacité de calcul. Cet inconvénient a été intégré ici par une hiérarchie des différents niveaux visualisés d'une part, et par un test sur un SdS plus important d'autres part.
Il y a au total 8 réseaux de Pétri différents répartis sur trois niveaux :
Réseaux de Pétri N0 :

- Architecture centralisée
- Architecture distribuée

Réseaux de Pétri N1 :
- Char
- Drone

Réseaux de Pétri N2 :
- Antenne
- Radio
- Serveur de communication
- Système d'information

1) Paramétrisation du réseau

Un réseau de Pétri utilise des probabilités et des délais spécifiques pour chaque transition, qui déterminent la probabilité de franchissement de cette transition par rapport à d'autres. Dans notre modélisation, nous avons cherché à rendre les probabilités et fréquences facilement adaptables pour prendre en compte les probabilités et les fréquences d'événements réels. Les valeurs ont été choisies cohérentes de la réalité, mais pour des raisons de confidentialité, aucune valeur de système en service n'a été fourni.
Une loi exponentielle modélise la durée de vie d'un phénomène sans vieillissement, ni usure. Aussi, la durée de vie des composants des systèmes est modélisée par une loi exponentielle. Le temps d'envoi en réparation de composants et le temps de création d'un message sont modélisés par des lois uniformes. Les délais sont modélisés par des lois de Dirac et enfin des tirages sont effectués lors de la détermination de réussite ou d'échec des réparations ainsi que pour le choix du destinataire du message.
Plusieurs hypothèses ont été formulées afin de simplifier la modélisation, voici nos hypothèses :
- Hypothèse 1 : Architecture centralisée → le char commande reçoit un message et décide s'il veut le transmettre à un autre ou pas, sous forme d'un nouveau message.
- Hypothèse 2 : Les drones peuvent être réparés en cours de mission.
- Hypothèse 3 : Un seul jeton au niveau N2, donc les événements ne peuvent pas arriver tant que le précédent n'est pas résolu.
- Hypothèse 4 : Deux messages ne peuvent pas se croiser (délai de transmission suffisamment court).
- Hypothèse 5 : Il y a toujours quelqu'un de disponible pour réparer, selon la loi uniforme choisie.
- Hypothèse 6 : Si un binôme Antenne/Radio ne fonctionne plus, l'autre est utilisé. Si aucun des deux ne fonctionne, le message est perdu.

2) Construction du réseau de Pétri N2 - Composant

Tous les composants fonctionnent de la même manière, il y a un total de cinq branches par composant, chacune représentant un événement :
- Panne
- Dommage
- Bug
- Ecoute
- Intrusion

Les événements « panne », « dommage » et « bug » sont des cas étudiés dans le cadre de l'événement redouté 1. L'événement « écoute » représente l'événement redouté 2 et l'événement « intrusion » représente l'événement redouté 3. D'autres événements peuvent être ajoutés sur le même modèle. (Fig. 5 et 6)

En utilisant GRIF/Petri, une transition avec un délai distribué exponentiellement est représentée en blanc, une transition sans délai est représentée par un fin rectangle noir, et tout autre transition déterministe est représentée par un large rectangle noir. Pour les délais de transition, les lois disponibles dans GRIF/Petri sont les lois : exponentielle, uniforme, Dirac, triangulaire, Weibull, normale, et log-normale.

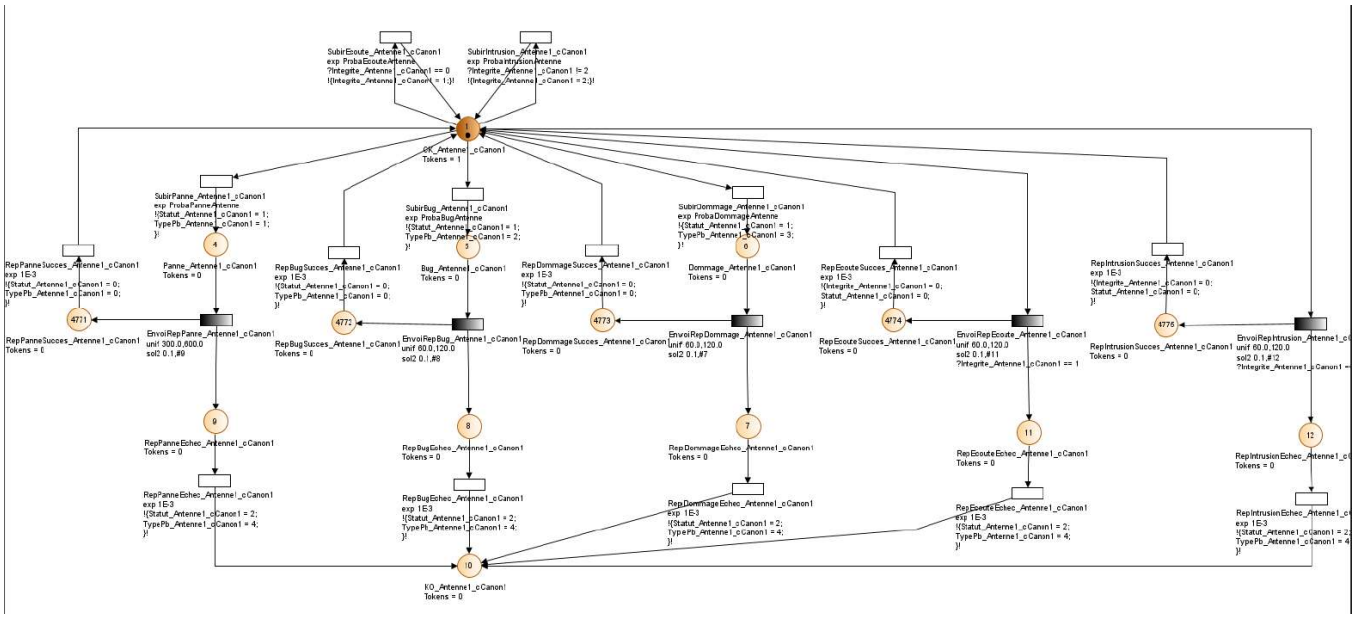

Fig. 5. Fonctionnement du réseau de Pétri du composant

La place dans laquelle le jeton est présent représente l'état actuel du composant. En suivant une loi exponentielle, le jeton peut arriver dans une place "dysfonctionnement". Par la suite, une loi uniforme déterminera le délai de réparation du composant et un tirage décidera si la réparation est un succès ou un échec. En cas de réussite, le jeton retournera dans sa place d'origine et en cas d'échec le jeton ira dans la place "KO" signifiant que le composant est hors-service.

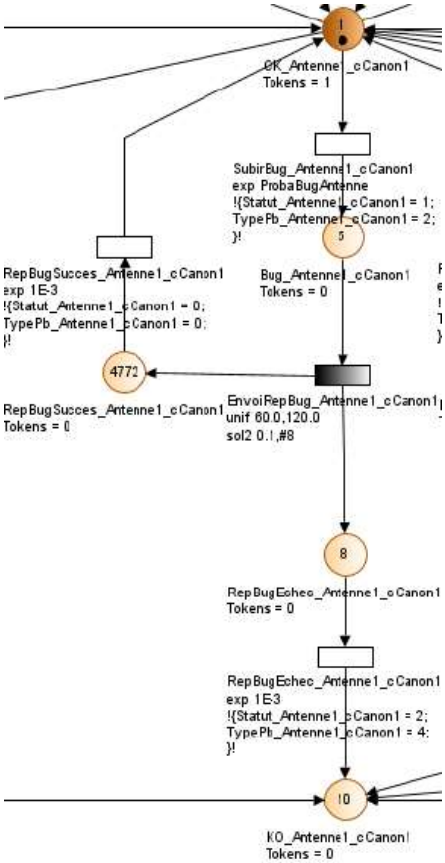

Fig. 6. Détail du fonctionnement du réseau de Pétri du composant

3) Construction du réseau de Pétri N1 – Char et Drone

Les chars canons et le char commande sont modélisés par le même réseau de Pétri, c'est leur position dans l'architecture qui déterminera leur rôle. Le rôle de ce réseau de Pétri est double, il assure l'émission et la réception des messages par le véhicule.

La réception est modélisée par la partie de gauche. Lorsque le véhicule reçoit un message, un jeton (représentant le message) apparaît dans la place en haut à gauche. Ensuite, avec des transitions instantanées et des systèmes de gardes sur les transitions, l'état de chaque composant est vérifié. Si lors d'une vérification il apparaît qu'un composant est indisponible ou hors-service, le message est supprimé et n'est donc pas traité par le véhicule. Des compteurs sous formes de variables permettent connaître le nombre et la cause des messages perdus.
L'émission est modélisée fonctionne de la même manière que la réception. Un message est généré en suivant une loi uniforme, puis une vérification est effectuée sur chaque composant pour déterminer leur disponibilité. En cas de non-disponibilité, le message est supprimé et n'est donc pas émis.
Pour l'émission ainsi que pour la réception, si un membre du binôme antenne/radio est indisponible, alors le second binôme sera utilisé. Le message est supprimé uniquement si les deux binômes sont indisponibles.

Le réseau de Pétri du drone fonctionne de la même manière que le réseau de Pétri du char, à la différence près que le drone ne dispose pas de radio et d'une seule antenne. Il n'est pas répété ici par soucis de clarté.

4) Construction du réseau de Pétri N0 – Architecture distribuée

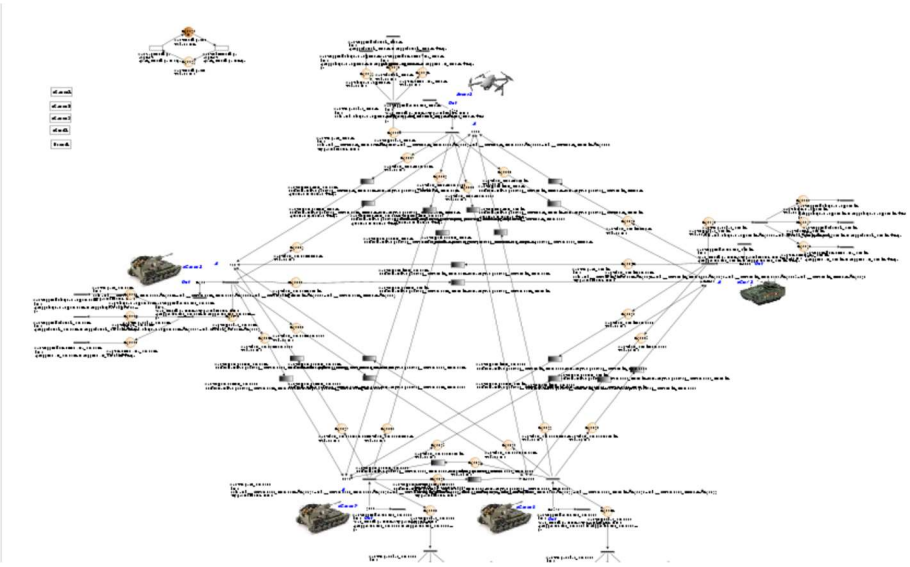

Fig. 7. Réseau de Pétri de l'architecture distribuée

Le réseau de Pétri de l'AR2 (architecture distribuée) est composé de quatre réseaux de Pétri « Char » ainsi que d'un réseau de Pétri « Drone ». D'autres véhicules peuvent être ajoutés, pour cela il faut relier le nouveau véhicule à tous les autres véhicules comme ceux déjà présents.
Lors de l'émission d'un message par un véhicule, un tirage détermine le destinataire du message. Il faut prendre garde lors de l'ajout d'un nouveau véhicule à adapter le tirage. Il y a également un risque que le message se perde en cours de chemin, ce qui est représenté par les événements « erreur humaine », « élongation », « obstacle » qui font partie des événements redoutés 1. Il est également possible d'ajouter un événement redouté lié au brouillage des communications, un système de brouillage simple a été intégré (en haut à gauche du réseau de Pétri) mais n'a pas été utilisé. (Fig. 7)

5) Construction du réseau de Pétri N0 – Architecture centralisée

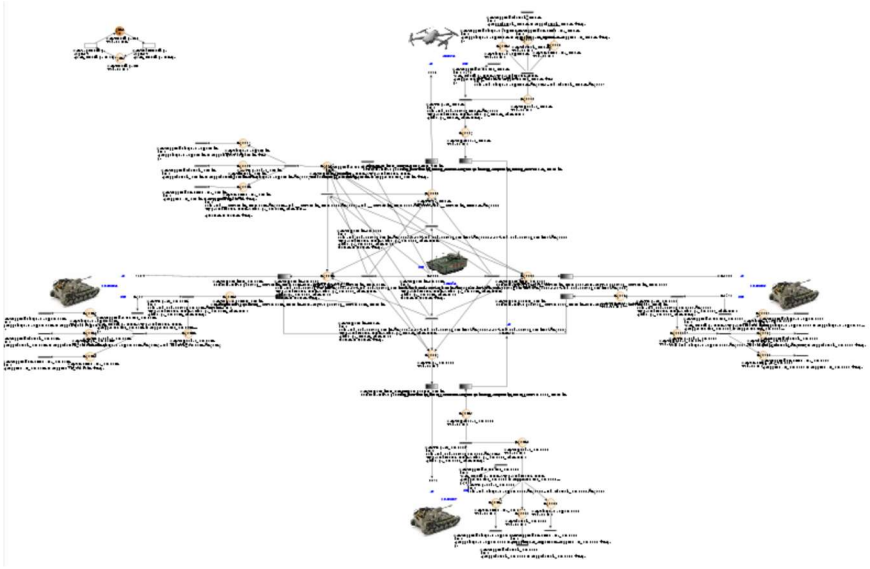

Fig. 8. Réseau de Pétri de l'architecture centralisée

De la même manière que pour le réseau de Pétri de l'AR2, celui de l'AR1 (architecture centralisée) est composé de quatre réseaux de Pétri "Char" ainsi que d'un réseau de Pétri "Drone". D'autres véhicules peuvent être ajoutés en reliant le nouveau véhicule au char commande de la même manière que les autres véhicules. Le char commande dispose d'un tirage pour sélectionner le destinataire du message ainsi que pour subir un événement redouté. Les autres véhicules eux n'ont qu'un tirage déterminant le succès ou non de l'émission du message. Le réseau de Pétri est conçu pour qu'un véhicule ne puisse pas s'envoyer de message à lui-même en prenant le char commande pour relais. (Fig. 8)

IV. RESULTATS

Les études exploratoires ont été réalisées sur un petit système de systèmes terrestres de 5 pions (3 chars canons, 1 char contrôle commande et 1 drone), restreint à trois événements redoutés sur les communications. Deux architectures ont été comparées : architecture distribuée ou architecture centralisée. De même, pour restreindre le périmètre, tout en restant représentatif, seuls quelques résultats sont évoqués dans la suite de ce document.

A. ER1 : Perte de Communication

Percent_ER1 : Pourcentage des messages perdus à cause de l'ER1, « perte des communication ». La courbe est fonction du temps.

ANALYSE

Percent_ER1 atteint 94.734% au maximum, et une moyenne de 87.9% pour l'architecture centralisée (Fig.9), là où elle atteint au maximum 66.634% et en moyenne 62.145% pour l'architecture distribuée (Fig.10). L'architecture distribuée permet donc une meilleure robustesse face à une combinaison de défaillances, des dommages ou bug.

Remarque : Le pourcentage est calculé par nombre de messages perdus à cause d'ER1 divisé par nombre de messages envoyés, chacune de ces valeurs étant prise à l'instant t, d'où la possibilité que la courbe soit décroissante.

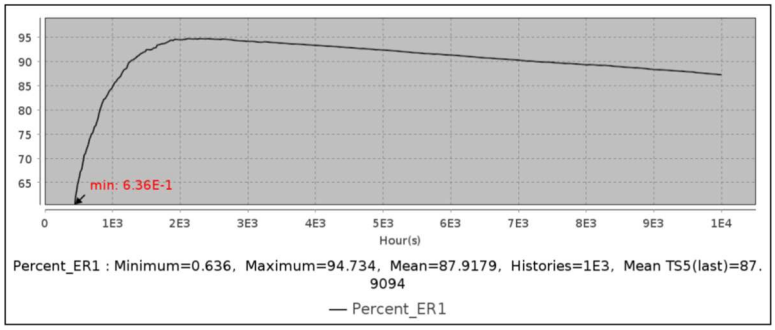

Fig. 9. Architecture Centralisée – Résultats ER1

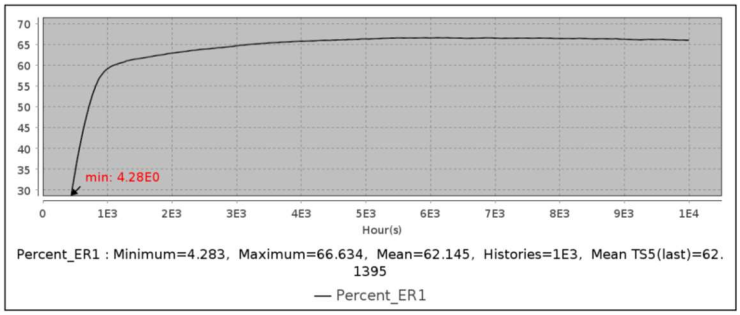

Fig. 10. Architecture Distribuée – Résultats ER1

B. ER2 : Interception des Communications

Percent_ER2 : Pourcentage des messages interceptés à cause de l'ER2, interception des communications (écoute). La courbe est toujours fonction du temps.

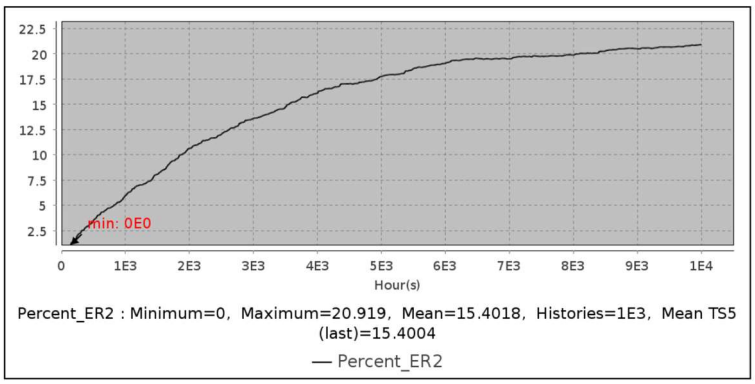

Fig. 11. Architecture Centralisée – Résultats ER2

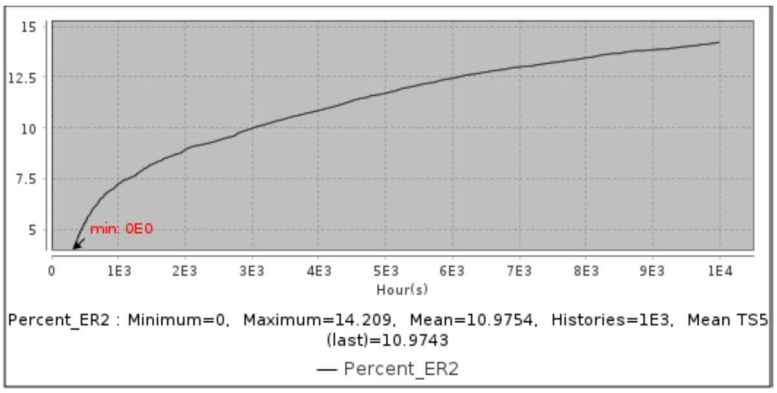

Fig. 12. Architecture Distribuée – Résultats ER2

ANALYSE

Percent_ER2 atteint 20.919% au maximum, et une moyenne de 15.4% pour l'architecture centralisée (Fig.11), là où elle atteint au maximum 14.209% et en moyenne 10.98% pour l'architecture distribuée (Fig.12). L'architecture distribuée permet donc une meilleure résilience face à une attaque de type écoute.

1) ER3 : Intrusion dans les systèmes de communication à des fins de désinformation

Percent_ER3 : Pourcentage des messages émis par un char qui est sous contrôle ennemi à l'instant t. La courbe est toujours fonction du temps.

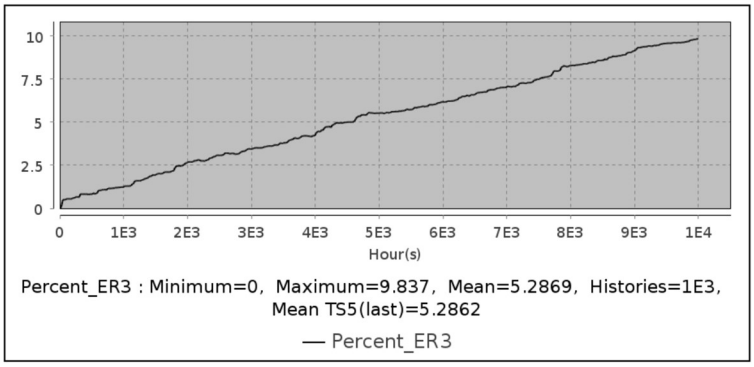

Fig. 13. Architecture Centralisée – Résultats ER3

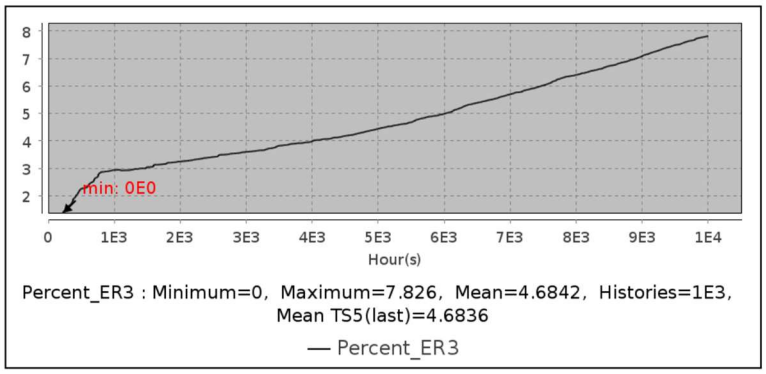

Fig. 14. Architecture Distribuée – Résultats ER3

ANALYSE

Percent_ER3 atteint 9.837% au maximum, et une moyenne de 5.29% pour l'architecture centralisée (Fig.13), là où elle atteint au maximum 7.826% et en moyenne 4.684% pour l'architecture distribuée (Fig.14). L'architecture distribuée permet donc une meilleure résilience face à une attaque de type désinformation.

C. Résultats Généraux:

La résilience d'une architecture face aux attaques combinées cyber-physiques dépend fortement de sa conception et de ses mécanismes de sécurité. Une architecture centralisée peut être plus facile à gérer et à sécuriser globalement, mais elle est plus vulnérable aux attaques ciblant le point central. À l'inverse, une architecture distribuée peut offrir une meilleure résilience en termes de redondance et d'absence de point de défaillance unique, mais elle nécessite des stratégies de gestion et de sécurité plus complexes pour assurer une protection adéquate.

Le choix entre une architecture centralisée et distribuée doit être basé sur une évaluation approfondie des besoins spécifiques du système, des ressources disponibles pour la sécurité, et du niveau de tolérance aux risques de l'organisation. L'analyse des trois événements redoutés liés à la survivabilité du SdS étudié ici est favorable à l'architecture distribuée, face à chacun des trois évènements redoutés étudiés, avec les paramètres choisis.

V. DISCUSSION ET PERSPECTIVES

Dans ce projet, deux architectures ont été comparées, l'architecture centralisée et l'architecture distribuée, soumis à trois événements redoutés. Pour effectuer cette comparaison, les réseaux de Pétri stochastiques hiérarchiques ont été utilisés afin de pouvoir composer facilement des systèmes de systèmes avec en tête l'idée de passer à des architectures plus complexes, jusqu'à 500 systèmes. Le processus stochastique permet d'effectuer des simulations et des analyses temporelles.

Nous avons observé diverses variables que nous avons jugé pertinentes afin de dresser des graphes permettant la comparaison entre les deux architectures. La principale conclusion de l'étude est donc que le réseau de Pétri est un outil intéressant pour évaluer la survivabilité des SdS, pour apporter des éléments de réponse.

Notre étude nous a permis d'identifier que l'architecture 2 (distribuée) est la plus robuste, face aux 3 ER. Mais on se limite ici à la communication, et à un seul jeu de données. De la même manière, la variation des différents paramètres pourrait influer

sur le résultat final, ou encore la constitution du système de systèmes. Il faudrait combiner d'autres aspects d'un véhicule terrestre afin de compléter cette étude et de pouvoir conclure réellement sur la comparaison des deux architectures.
Cette étude peut être améliorée de plusieurs manières avec notamment l'étude d'une architecture mixte entre l'architecture distribuée et l'architecture centralisée afin de déterminer une architecture optimale, compte tenu d'autres critères.
Enfin, le passage à l'échelle de 500 systèmes paraît possible, mais n'a pas encore été réalisé, représentant le nombre de pions d'un Groupement Tactique Inter-Armes, ce qui était également un attendu de l'étude de cas.
Les limites des outils issues de la recherche nous ont amené à tester un outil commercialisé, avec des calculs plus puissants permettant des premiers résultats et une conclusion.

VI. CONCLUSION

La survivabilité des SdS est un sujet émergent. Sur un exemple concret, l'étude a permis de vérifier l'utilité de ce type d'outil pour le domaine de la communication, et de tester différents outils pratiques, libres ou du commerce, sur la réalisation et l'analyse des réseaux de Pétri. L'étude d'une architecture mixte entre l'architecture distribuée et l'architecture centralisée afin de déterminer une architecture optimale d'un point de vue de la survivabilité, confronté à d'autres critères programmatiques sera le prochain point à étudier, ainsi que le passage à un plus grand nombre de systèmes constituant le SdS.

REMERCIEMENTS

Aux étudiants qui ont participés aux différents travaux d'exploration des réseaux de Pétri autour de ce sujet : Loïc de Haro, Constance Gaudin, Jean-Yves Combles, Mohammed Ez-Zraidy, Adam Bendou, Morgane Mallet, Paul Martinez, François Coustau-Guilhou, Quentin Logerais, Mathis Harouard, et Aude Bakayoko. Remerciements également à Salah Sadou pour le tutorat des étudiants de l'ENSIBS, à Guillaume Massonet et Naly Rakoto pour le tutorat des étudiants de l'IMT Nantes Atlantique, à Cyrille Folleau pour le prêt d'une licence étudiante Petri Grif, ainsi qu'à Françoise Cadoret et François Clerbout pour leurs soutiens et relectures.

REFERENCES

[1] SIGNORET, J.P. (2008) Analyse des risques des systèmes dynamiques : réseaux de Petri - Exemples de modélisation, *Techniques de l'ingénieur*
[2] WOODARD, M.J. (2017) Survivability modeling for cyber-physical systems subject to data corruption
[3] DINGLE, N., KNOTTENBELT, W., SUTO, T. (2009) PIPE2: A Tool for the Performance Evaluation of Generalised Stochastic Petri Nets
[4] DGA (2008) Guide S-CAT n°10032 Guide de prise en compte de la survivabilité des systèmes de systèmes (non publié)
[5] Durand, G., Lagorce, F., Desroches, V., (2012) Maîtrise de la survivabilité d'un système de systèmes, Congrés lambda mu 18 Tours
[6] Leboisselier, T., (2020) "Simulation numérique en aide aux études de survivabilité des systèmes de systèmes", Congrés lambda mu 22 Le Havre