Analyse et diagnostic des fautes dans un environnement multi-domaines : une approche hiérarchique et distribuée

Guiagoussou Mahamat, Anindya Das

Université de Montréal, Département
d'informatique et de recherche opérationnelle,
C.P. 6128, Succursale CENTRE-VILLE,
Montréal(Québec), Canada, H3C-3J7

Résumé : *Dans cet article, nous décrivons une approche hiérarchique et distribuée d'analyse et de diagnostic des fautes dans un environnement multi-domaines. L'approche est basée sur les modèles normalisés de gestion de réseaux et les idées développées par l'intelligence artificielle distribuée selon lesquelles un ensemble d'agents intelligents coopèrent pour la résolution d'un problème. L'article donne dans un premier temps une description sommaire des modèles de base utilisés. Il présente ensuite l'approche d'analyse et de diagnostic des fautes. La conclusion du document présente nos futurs travaux.*
Mots-clés : *Gestion de réseaux, diagnostic de fautes, modèle de réseau.*

1. Introduction

Les réseaux de demain offriront à leurs utilisateurs des facilités de transfert et de traitement de tous les types d'informations: données, voie, graphiques, images. Des applications diverses telles que les conférences multimédia, la distribution des vidéos vont alors émerger. Ces applications manipulent des gros volumes de données. Certaines d'entre elles requièrent en plus une garantie des bonnes qualités des services de base. Aussi les performances des ressources matérielles disponibles (ordinateurs, stations de travail, serveurs de fichiers, multiprocesseurs, etc.) sont de plus en plus grandes. Les coûts des lignes à haut débit sont relativement bas au niveau local.
La garantie des bonnes qualités des services offerts par les réseaux dépend largement de l'efficacité de la gestion globale de l'environnement résultant. Cependant la complexité de cet environnement rend la gestion de l'ensemble très difficile. L'hétérogénéité et la distribution en sont les deux principaux facteurs. Dans le cas de la maintenance des réseaux, les activités réalisées nécessitent des procédures très complexes et trop longues. Elles affectent plusieurs sous-systèmes du réseau. Les informations pertinentes à la localisation des fautes sont incomplètes et souvent incertaines. Il y a un manque d'outils appropriés pour la collecte et l'analyse d'informations sur les fautes.
Dans le cadre nos travaux de recherche, nous considérons un réseau comme un ensemble de domaines interconnectés. Nous étendons ce modèle de réseau aux idées développées par l'intelligence artificielle distribuée. Cet article propose une approche hiérarchique et distribuée pour l'analyse et le diagnostic des fautes dans un environnement multi-domaines. L'article donne dans un premier temps une description sommaire des modèles de base utilisés. Il présente

ensuite l'approche de diagnostic hiérarchique et distribué des fautes. La conclusion du document présente nos futurs travaux.

2. Modèles de base

Dans une étude de l'état de l'art en gestion des réseaux [3], nous avons donné les motivations d'un modèle simple et flexible pour une gestion globale et efficace des réseaux. La nécessité d'un modèle fonctionnel indépendant de la technologie ressort dans plusieurs discussions [1,4].

2.1 Modèles normalisés

Les modèles normalisés (ISO 10040, CCITT X.701) considèrent un réseau comme un domaine décomposable en d'autres domaines fonctionnels de gestion (management domain). Chaque domaine définit une frontière à l'intérieure de laquelle un certain nombre de ressources (managed objects) sont gérés par un processus de gestion-système (manager). Les différents gestionnaires exécutent leurs opérations de façon distribuée et coopérante. Ils communiquent entre eux au moyen des services offerts par les éléments de service application (ASE : Application Service Element) du protocole CMIP (Common Management Information Protocol). Un domaine possède en plus un ou plusieurs processus appelés agents. Chaque agent représente un certain nombre d'objets gérés. Un gestionnaire peut avoir accès à un objet géré via l'agent qui lui est relié.
Seules les interactions, entre et à l'intérieur des domaines fonctionnels sont normalisées. Des extensions à ce modèle sont donc nécessaires afin de permettre le développement de sous-systèmes de gestion de plus bas niveau.

2.2 Modèle de l'intelligence artificielle distribuée

L'intelligence artificielle distribuée (Distributed Artificial Intelligence : DAI) semble être une bonne candidate pour cette extension. Des schémas conformes à la DAI ont été appliqués avec succès dans certains travaux [2, 4]. Des tels schémas offrent un cadre de formalisation d'un réseau par un système où les connaissances sont distribuées. Le réseau à gérer est considéré comme un domaine de gestion décomposable en un ensemble de domaines plus restreints et plus maîtrisables. Chaque domaine peut être à son tour récursivement défini comme une interconnexion de domaines. Ce processus de décomposition récursif peut être ainsi répété à tout niveau jusqu'à l'obtention de domaines dont le contrôle et l'administration ne relèvera que d'un seul agent de gestion. Un ensemble d'entités intelligents dits agents coopérent au sein d'un système de gestion multi-agents pour résoudre un problème de gestion de réseau. Chaque agent est spécialisé dans une tâche spécifique de gestion des réseaux et sa connaissance est représentée par un modèle. Chaque processus utilise une connaissance locale et communique avec des hôtes distants. Dans le domaine de la DAI, les recherches sont orientées vers la conception d'un schéma d'organisation distribué permettant la résolution des problèmes. Cette organisation doit décrire les mécanismes utilisés pour faire coopérer les agents.

2.3 Modèles hiérarchiques

Nous proposons une extension aux modèles précédents qui consiste à représenter les connaissances sur la configuration réelle du réseau par la hiérarchie de ses composantes (topologie et composition internes).

2.3.1 Modèle de réseau

La figure 1(a) décrit la notation adoptée le long de ce document pour représenter un réseau. La figure 1(b) donne un exemple de réseau composé de trois noeuds simples, de trois lignes de communication et d'un sous-domaine. Le sous-domaine représenté par son gestionnaire local g1 est vu par le gestionnaire principal (master manager) comme un noeud complexe. Cette abstraction permet au gestionnaire principal g d'avoir une vue simplifiée, mais complète du réseau. Ce qui offre un cadre de gestion globale au moyen d'une coopération entre les gestionnaires locaux d'une part et entre les gestionnaires locaux et le gestionnaire principal d'autre part. Chaque gestionnaire local est en charge de la gestion de son sous-domaine, alors que le gestionnaire principal est en charge de la gestion globale du réseau.
La figure 2 présente une vue détaillée du sous-domaine 1 qui est représenté par un sous-graphe du domaine principal. Cette vue est une vue restreinte au gestionnaire g1. Dans notre exemple, g1 joue aussi bien le rôle de gestionnaire du sous-domaine 1, que celui d'agent de surveillance du noeud 1. La connexion entre le noeud 2 et le sous-domaine 1 (noeud complexe g1) se fait à travers le noeud 1.1. Cette connaissance est détenue par les deux gestionnaires (g et g1). Les échanges d'informations de gestion se font entre les agents de gestion (gestionnaires, agents de surveillance) d'une part, et entre les agents de surveillance et les objets gérés d'autres parts.
La hiérarchie des configurations dérivée d'un réseau multi-domaines est une arborescence qui décrit la topologie d'interconnexion et la structure interne des composantes du réseau. La figure 3 décrit une vue hiérarchique des configurations des domaines du réseau de la figure 1(b). Les composantes abstraites du réseau sont des noeuds, des lignes, etc. Un noeud peut être composé de carte CPU, de carte horloge, de carte de port, etc. Une application utilise les ressources d'un noeud (temps CPU, espace mémoire, etc.) et assure un service donné. Les lignes de transmission peuvent être composées d'équipements de satellite, de câble torsadée, ou de la fibre optique. Ce sont des ressources porteuses ne contenant pas d'autres lignes. Par contre plusieurs canaux peuvent être dérivés d'une même ligne. Des sous-canaux peuvent être à leur tour dérivés de ces canaux (un canal 8 bits/s peut porter jusqu'à 4 canaux de 2 bit/s).
Plusieurs vues hiérarchiques d'un réseau peuvent être dérivées à partir de sa configuration réelle. Chaque vue vise un objectif donné lié à l'analyse d'un problème de diagnostic (plaintes des usagers, dégradation d'un service, isolement d'une partie du réseau, trafic congestionné, propagation de faute dans un domaine, défaillance d'un objet géré, etc.).

FIGURE 1: Modèle abstrait de réseau et notations associées

a) **Notations :** G = (S, A, g) : réseau R géré par un gestionnaire g

. S = [s / s est un nœud de G géré par g]

. A = [aij / i, j dans S et aij lie i et j]

Nœud simple: interactions avec nœuds adjacents et gérant local. Possibilité de décomposition interne

Nœud complexe: gestion de tous les objets du domaine, relations avec nœuds adjacents (simples et complexes), et interactions avec gérant de niveau supérieur

Agent: intermédiaire entre nœud complexe (gestionnaire), nœuds simples et lignes associées

Gestionnaire principal (Master manager): gestion globale du réseau

Échanges d'informations de gestion

Ligne de communication (objet géré entre les nœuds)

b) **Exemple :**

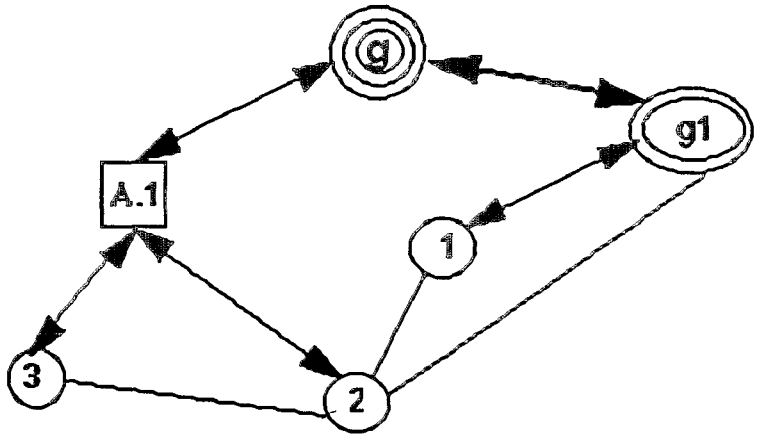
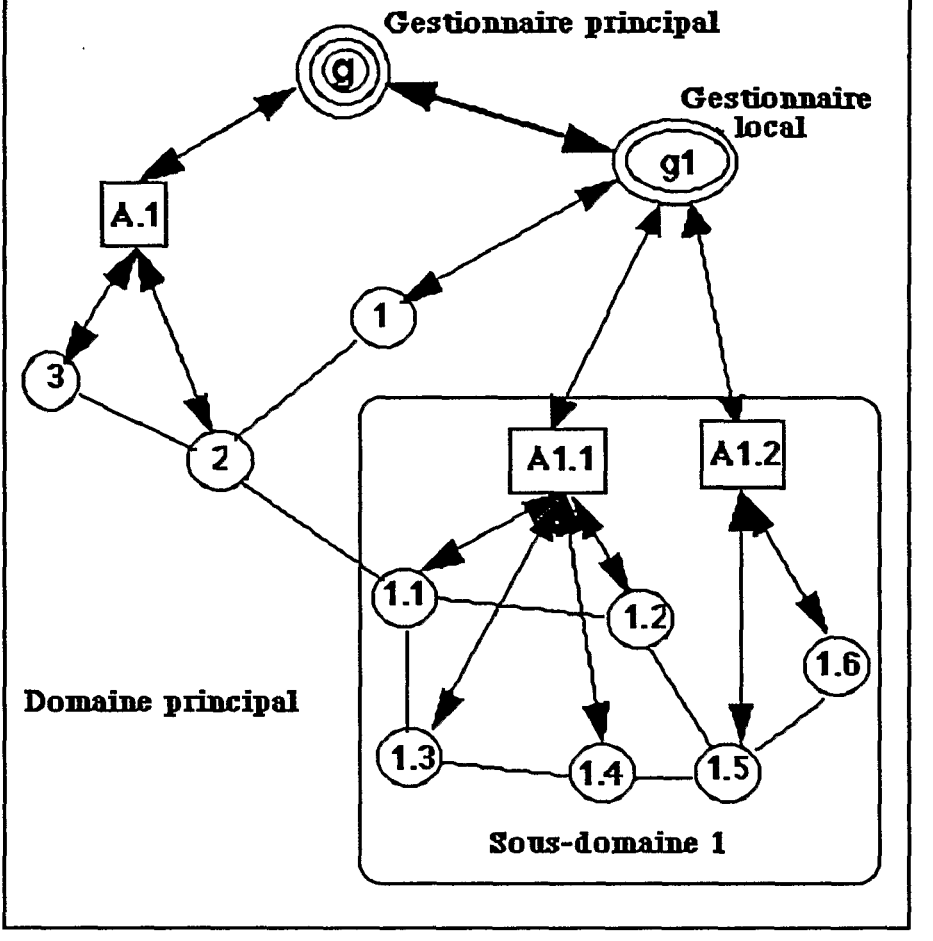


FIGURE 2 : Description détaillée du réseau de la figure 1

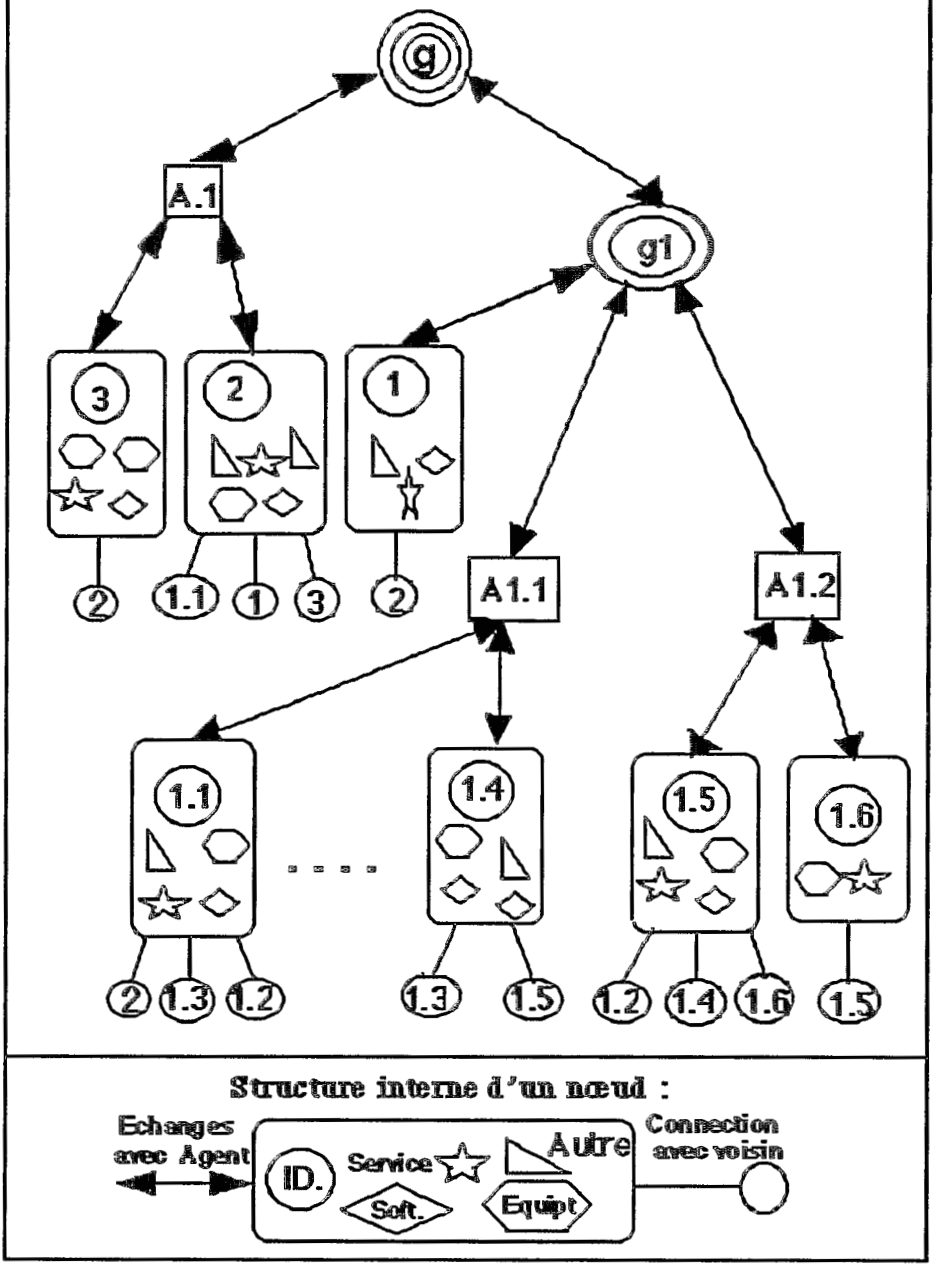


Chaque gestionnaire analyse une vue du sous-arbre décrivant la configuration interne de son domaine pour reconnaître un motif de propagation d'une faute, détecter un délai anormal ou un isolement d'une partie du domaine. Les feuilles du sous-arbre représentent les composantes au niveau desquelles s'arrête les opérations d'analyse et de localisation des fautes.
Une discussion un peu plus détaillée sur l'utilisation des vues pour le diagnostic de fautes sera donnée dans les prochaines sections.

2.3.2 Modèle hiérarchique des fautes

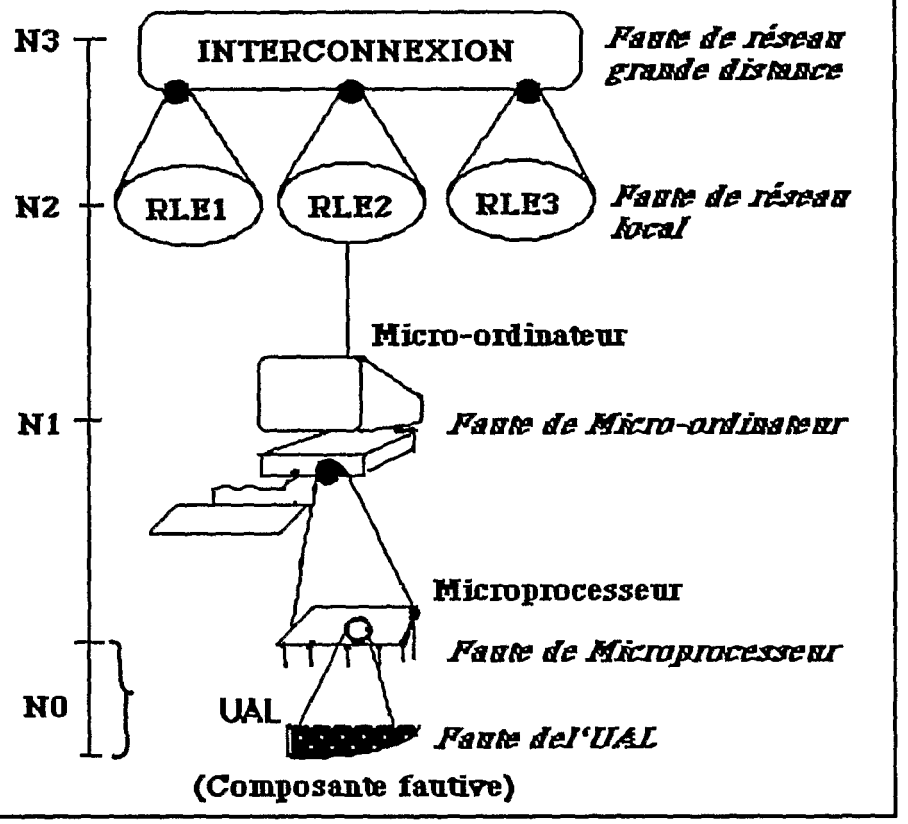
La modélisation par domaine permet de définir une hiérarchie des fautes de réseaux [3]. Un réseau peut être considéré à plusieurs niveaux d'abstraction. Le plus haut niveau indique le niveau le plus abstrait ou se trouve le centre de gestion du réseau. On y trouve la gestion globale du réseau (master manager). Les plus bas niveaux indiquent les feuilles de la hiérarchie des composantes.

FIGURE 3 : Vue hiérarchique du réseau de la figure 2



La figure 4 décrit un modèle hiérarchique de fautes pour un domaine obtenu par l'interconnexion de trois réseaux locaux (RLE1, RLE2 et RLE3).

FIGURE 4 : Modèle hiérarchique de fautes réseaux



Toute faute détectée au niveau du grand réseau (centre de contrôle, usagers, etc.) peut avoir des causes distantes. A ce haut niveau on parlera de *faute de réseau grande distance* (localisation abstraite). La faute peut avoir son origine aussi bien dans le RLE1, dans le RLE2 que dans le RLE3. Elle peut aussi avoir son origine dans les interactions entre les trois RLEs. Supposons qu'une des composantes du RLE2 est la cause de la faute. Des résultats préliminaires peuvent être obtenus déclarant une faute au niveau réseau local. Il s'agit d'une faute de niveau inférieur a celle décrite précédemment. On peut parler de *faute de réseau local.* Il faut d'avantage descendre dans la hiérarchie des composantes pour déterminer la composante à réparer ou à remplacer. Supposons que la composante fautive du RLE2 est un micro-ordinateur. Un nouveau niveau de la hiérarchie peut être défini : *faute de micro-ordinateur.* Le dernier niveau est celui de la composante du micro-ordinateur qui a causé la faute. Si cette composante est le microprocesseur, on définira une *faute de microprocesseur.* Dès ce niveau, on peut arrêter le diagnostic et remplacer le microprocesseur. Mais si grâce au VLSI, on peut remplacer ou réparer l'unité arithmétique et logique qui aurait causé la panne du microprocesseur, la définition d'un dernier niveau (*faute d'UAL*) reste justifiée.

3. Diagnostic hiérarchique et distribué

Avant de décrire les différentes étapes de l'approche de diagnostic, nous donnons une liste des besoins que doit satisfaire un système de diagnostic de fautes. Dans cette section, nous donnons aussi un exemple d'application de l'approche hiérarchique et distribuée pour le diagnostic des fautes permanentes.

3.1 Besoins à satisfaire par un système de diagnostic

Les systèmes de diagnostic de fautes doivent répondre à plusieurs attentes qui leurs sont actuellement adressées [1]. Ils doivent pouvoir traiter des flots d'alarmes asynchrones aussi rapidement que la vitesse moyenne à laquelle elles sont générées. La maintenance d'un modèle précis de la configuration du réseau est très pertinente (topologie du réseau à tous les niveaux, configuration des équipements). Il faut aussi tenir compte des propriétés changeantes des fautes. La sévérité des fautes est souvent fonction du trafic, de la journée, de la semaine, etc. Une autre fonction à satisfaire est la séparation du problème principal des effets de bord. L'interconnexion des réseaux provoque en effet la propagation des fautes et conduit à plusieurs manifestations différentes de la même faute. Le système doit pouvoir traiter les problèmes par ordre de sévérité. Lors des attentes des résultats de tests le système doit pouvoir remplacer le problème courant par le problème le plus sévère, tout en évitant les tests destructifs pendant les périodes de pointe. Il faut aussi déterminer et utiliser des probabilités variables des composantes suspectes. Les tests peu coûteux et qui donnent plus d'informations doivent être privilégiés. L'automatisation des tests permet de réduire le recourt à l'assistance humaine. Elle permet aussi de sélectionner et d'implémenter rien que des tests autorisés. Enfin, le système doit pouvoir interpréter les résultats de tests et effectuer des diagnostics corrects.

3.2 Approche de diagnostic

L'approche débute par une étape d'initialisation. La détection des anomalies est une opération permanente assurée par les agents de surveillance. Les autres opérations qui attirent notre attention sont essentiellement le filtrage, la corrélation des alarmes et les tests de diagnostic. L'approche est basé sur le parcourt de l'arbre de la hiérarchie des composantes du réseau. Ces parcours offrent un cadre approprié pour la corrélation des alarmes et facilitent la reconnaissance (1) des motifs précis de propagation des fautes, (2) des délais anormaux, et (3) de l'isolement d'une partie du réseau.

3.2.1 Initialisation

Initialement une description de la hiérarchie des composantes permet de définir les niveaux d'abstraction, les différents objets impliqués et leurs interrelations. Chaque niveau de la hiérarchie indique un niveau d'abstraction et permet de considérer des unités locales d'analyse et de diagnostic de fautes (gestionnaire) des domaines du réseau. Chaque domaine de gestion se compose d'un nombre donné d'éléments physiques et logiques. A cette étape, l'attribut "état de santé" de chacun de ces éléments est mis à une valeur initiale ("Normal", 1.0, 100%,

etc.) qui représente un état opérationnel satisfaisant. Toute probabilité de ces éléments d'être à l'origine d'une faute quelconque est fixée à zéro. On suppose à ce niveau initial que le réseau ne contient aucune faute.

3.2.2 Détection des anomalies

Au premier niveau de la hiérarchie se situe la surveillance du réseau qui permet de détecter les anomalies au moyen des senseurs, dispositifs, ou agents de surveillance. Le domaine de contrôle est le réseau tout entier. Les sous-domaines sont vues comme des composantes à gérer (noeuds complexes). Un processus agent de détection peut être défini pour les tâches de collecte d'informations liées à la détection des fautes par des "polling". Les mécanismes classiques de tests tels que les "ping" peuvent aussi être utilisés. Les signaux provenant des capteurs ou des composantes du réseau sont convertis dans un format commun de notification (normalisé) puis envoyés aux gestionnaires et agents concernés. Les plaintes et autres manifestations des diverses insatisfactions provenant des usagers sont aussi traduites sous forme de notifications ou d'alarmes.

3.2.3 Opérations d'analyse et de diagnostic

Filtrage et corrélation des alarmes : Les premières tentatives de localisation des fautes sont effectuées après filtrages et corrélations des alarmes. Le filtrage permet de catégoriser les alarmes et d'en reconnaître des types bien définis. Il permet aussi de faire une classification des alarmes par ordre de priorité. La corrélation détecte par la suite les ressemblances entre les alarmes. Ces opérations de base se font à tous les niveaux hiérarchiques du réseau. Elles visent à diminuer au maximum possible le nombre d'alarmes redondantes. Ce qui réduit aussi le nombre de composantes suspectes. Après filtrage et corrélation, les alarmes non redondantes et prioritaires sont traitées immédiatement. La liste des problèmes (pannes) liés aux alarmes, et celle des composantes suspectes sont mises à jour. Un poids correspondant à la chance (probabilité) d'être à l'origine d'une faute est alloué à chaque composante suspecte. Les opérations qui vont suivre analyse les données reçues et mettent à jour ces poids. Au dépassement d'un seuil, ou à la reconnaissance d'un motif de faute, la composante sujette au débordement du seuil est déclarée fautive. La valeur de son attribut état de santé est mise à jour ("Anormal", 0.0, 0%, etc.). Dans le cas de dégradation progressive de l'état de la composante, une mise à jour incrémentale doit être faite ("A surveiller", 0.35, 35%, etc.).

Niveau N : Des tests sont effectués sur la liste des suspects restants afin de déterminer la ou les composantes effectivement fautives. Les résultats des tests sont analysés et des conclusions sont prises par le gestionnaire de niveau N. Si les résultats obtenus suffisent pour aboutir à un diagnostic de la faute, le gestionnaire principal en est immédiatement informé. Ce dernier vérifie si les conclusions proposées sont exactes et propose des réparations pour éviter des conséquences catastrophiques de la faute.

Niveau N-1 : Si les résultats obtenus par le gestionnaire de niveau N ne sont pas suffisants pour conclure le diagnostic, des tests supplémentaires au niveau

N-1 (sous-domaine) doivent être effectués. Un parcourt descendant (top-down) de la hiérarchie des composantes est alors initié. Dans ce cas, le gestionnaire du niveau N détermine la branche qui présente la plus grande probabilité de contenir la faute et délègue le gestionnaire de niveau N-1 à poursuivre l'opération. Le parcourt se termine aux plus bas niveaux de la hiérarchie (feuilles de l'arborescence) où se trouvent les objets à réparer. A ce niveau le processus de diagnostic s'arrête, indiquant la ou les composantes fautives.

Un parcourt ascendant peut servir à répondre aux gestionnaires de niveau supérieur. Il permet aussi de faire une gestion préventive par blocage de la propagation des fautes réduisant ainsi les alarmes redondantes au niveau supérieur. Nous donnerons aussi un exemple de corrélation des alarmes réalisée au moyen d'un tel parcourt.

Coopération entre gestionnaires : Si des interactions entre les gestionnaires des domaines sont nécessaires pour terminer les opérations en cours, le gestionnaire principal du réseau peut servir d'intermédiaire. Il pourra se charger de gérer les conflits et d'éviter les restrictions éventuelles que peuvent rencontrer les différents gestionnaires impliqués. Il peut par exemple accéder à des informations sur un noeud distant et fournir à un gestionnaire local les informations demandées.

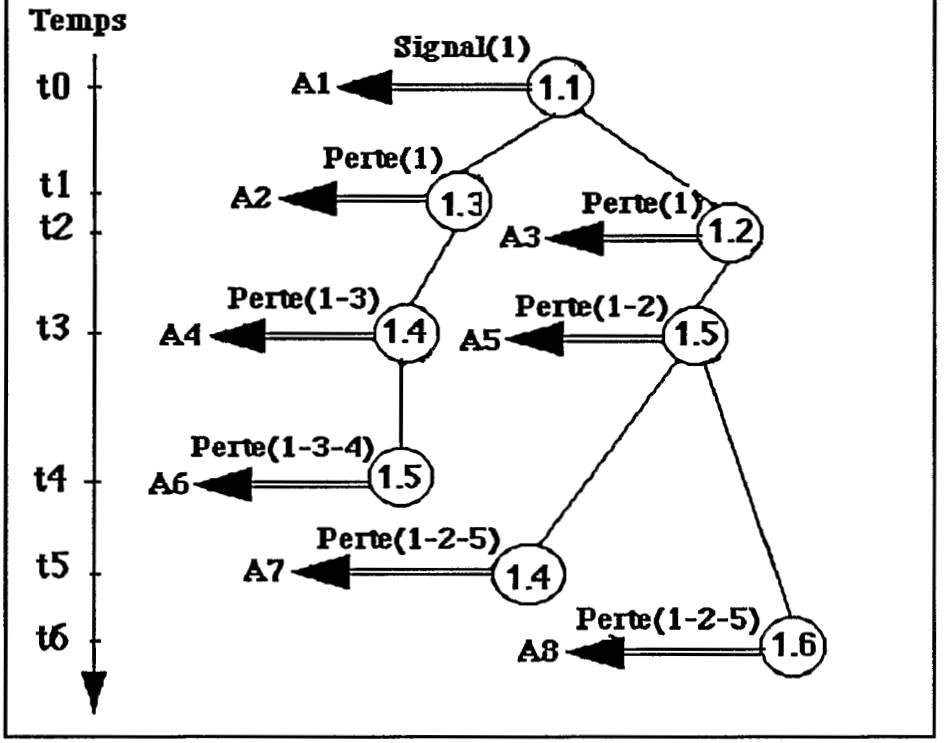
3.2.4 Parcourt de l'arbre de la hiérarchie des composantes

Au lieu d'écarter systématiquement les alarmes redondantes dans un domaine, on peut les utiliser comme complément d'informations. Une nouvelle alarme peut apporter une information pertinente portant sur une même composante : perte totale de synchronisation, niveau de dégradation du service, etc. Ce qui permet d'incrémenter la probabilité de cette composante à être la cause d'une ou de plusieurs fautes. Notre approche anticipe la propagation de la faute de proche en proche. Dès réception de la première alarme portant sur un problème donné, des tests sont effectués par les voisins immédiats (plus proches) de la composante suspecte, puis les prochains voisins immédiats, et ainsi de suite. Ces voisins sont supposés non fautifs. Toute réponse anormale de la composante suspecte est alors signalée au gestionnaire. Les alarmes reçues sont progressivement retracées (dans le temps) le long d'une vue hiérarchique du domaine. A la reconnaissance d'un motif précis de faute, ou au dépassement d'un seuil, les premières conclusions de diagnostic peuvent être proposées.

Motif de faute : Considérons l'avènement d'une faute permanente au niveau du noeud 1.1 (figure 2). La figure 5 décrit un motif de propagation de cette faute le long de la hiérarchie des composantes. En l'absence d'une autre faute, la même cascade d'alarmes (A_1, A_2, ..., A_8) sera reçue par le gestionnaire g1. Toutes ces alarmes se rapportent au noeud 1.1 du sous-domaine 1 ou à un chemin incluant le noeud 1. A une date t_0, la première alarme (signal(1)) est envoyée par le noeud lui même ou tout dispositif de détection d'anomalie associé à ce noeud. Les deux voisins immédiats (1.2 et 1.3) effectuent alors des tests pour s'assurer de l'état du noeud 1.1. Celui qui a le temps de réponse le plus faible, envoie à la date t_1 une alarme au gestionnaire pour signaler la perte de communication avec le noeud 1.1 (Perte(1)). Dans notre exemple, il s'agit de

1.3 qui sera suivi de 1.2. Les prochains voisins 1.4 et 1.5 vont signaler la perte de communication avec le noeud 1.1 via 1.3, et 1.2 (respectivement). Et ainsi de suite. Tout nouvelle alarme portant sur un problème du noeud 1.1, conduit à une incrémentation de son poids (probabilité de défaillance). Si le motif de la faute est détectable sans ambiguïté, le gestionnaire local propose immédiatement ses conclusions de diagnostic. Dans le cas contraire, le poids est comparé à un seuil fixé. Ce n'est qu'au dépassement de ce seuil que la composante sera déclarée fautive. Notons qu'au niveau du domaine principal, un motif de propagation de la même faute peut être parallèlement tracé. Le gestionnaire g1 informe le gestionnaire g qu'une opération de diagnostic d'une faute du noeud 1.1 est en cours. Ce dernier initialise des tests au niveau noeud 2, puis au niveau des noeuds 1, et 3 et analyse les résultats. Les résultats obtenus seront utilisés pour vérifier l'exactitude du diagnostic donné par g1 et les conclusions définitives du diagnostic au niveau global du réseau.

FIGURE 5 : Motif de propagation d'une faute; parcourt descendant (Faute du noeud 1.1, figure2)

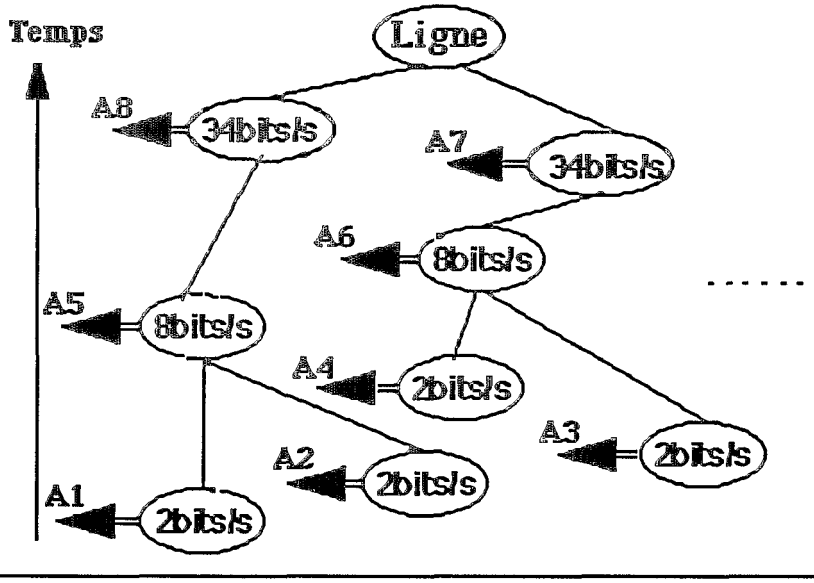


Délais anormaux et isolement d'une partie du réseau : Nous avons supposé dans le cas précèdent, qu'il n y a pas d'isolement d'une partie du domaine, ni de délais anormaux. Notre approche d'analyse permet de détecter aussi bien les délais anormaux que l'isolement d'une partie du domaine. Considérons par exemple un délai de réponse d très significatif sur la ligne (1.1,1.3); un délai

supérieur à celui de la ligne (1.1,1.2). Toutes les alarmes sur la branche 1.1-1.3 de la figure 5 seront retardées de d. Si les temps de réponse normaux sont estimés à l'avance par le gestionnaire, une comparaison des dates de réception des alarmes (t_1+d pour A_2, t_3+d pour A_4, et t_4+d pour A_5) aux dates attendues (t_1, t_3, et t_4) permet de détecter les écarts. Notons que si deux alarmes arrivent à la même date (Figure 5 : A_4, A_5), la plus prioritaire est considérée en premier lieu. L'isolement d'une partie du domaine se traduit simplement par l'absence d'un certain nombre d'alarmes. L'isolement du noeud 1.5 aura pour conséquence la perte des alarmes A_5, A_6, A_7, A_8.

Parcourt ascendant : On vient de voir comment à l'aide d'un parcourt descendant de l'arborescence, on peut reconnaître un motif précis de propagation d'une faute, détecter des délais anormaux et reconnaître l'isolement d'une partie du domaine. Avec un parcourt ascendant, on peut en plus déterminer des indices de corrélation des alarmes et aboutir à un diagnostic plus précis. La figure 6 présente un exemple de corrélation des alarmes au moyen d'un parcourt ascendant. La série d'alarmes se rapportant à l'ensemble des canaux supportés par une ligne défaillante est analysée pour fin de corrélation. On utilise un mécanisme de poids associé aux alarmes. Des réception d'une alarme de bas niveau (canaux 2bits/s) un poids fixe lui est associé. Tout canal 8 bits/s se fera ensuite attribuer la somme des poids des canaux 2bits/s qu'il supporte et qui ont généré des alarmes. De même le poids de tout canal 34 bit/s sera obtenu par la somme cumulée des poids des canaux 8bits/s qu'il supporte.

FIGURE 6 : Parcourt ascendant d'une faute; cas de la coorélation des alarmes (de canaux à ligne)



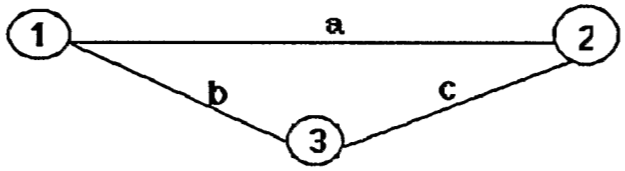
La procédure continue jusqu'au niveau de la ligne porteuse des canaux de plus haut niveau. Au dépassement d'un seuil de contrôle, la ligne sera alors déclarée "source principale" de la faute sous diagnostic.

Feuilles de l'arbre : Les feuilles de l'arbre correspondent aux composantes du réseau au niveau desquels aucun test de diagnostic supplémentaire n'est nécessaire pour localiser les fautes. Dans des cas non hiérarchiques, la logique du diagnostic hiérarchique peut toujours être conservée jusqu'à un niveau approprié défini au cours des opérations de diagnostic. A ce niveau, les composantes impliquées sont regroupées sous forme d'élément du réseau ou NE (Network Element) qui sera considéré fautif. Des diagnostics supplémentaires peuvent cependant être requis au niveau du NE.

3.3 Exemple de diagnostic des "Break-Faults"

Dans cette partie, nous décrivons l'organisation des opérations de diagnostic dans le cas particulier des fautes permanentes. Pour des raisons de simplicité, nous considérons le domaine restreint de la Figure 7.

FIGURE 7 : Exemple de domaine restreint D=(1,2,3,a,b,c)



3.3.1 Principe de diagnostic

Chaque gestionnaire local met à jour les paramètres de contrôle des objets qui sont sous sa responsabilité. A l'avènement d'un problème dans un domaine, les agents de surveillance et détection d'anomalies envoient une notification aux gestionnaires concernés. Afin de déterminer la composante qui aurait causé le problème détecté, tout gestionnaire local impliqué initialise les opérations de diagnostic. Ces opérations peuvent aussi être commandées par le gestionnaire de niveau supérieur. A un niveau local donné le diagnostic se fait par la mise à jour de plusieurs listes. La liste des alarmes (notée L(A)) est mise à jour par les opération de filtrage et de corrélation des alarmes et contient un historique de toutes les alarmes. La liste des problèmes (L(P)) contient les problèmes en cours et en attente de traitement. La liste des suspects (L(S)) contient l'identification des composantes du domaine ayant une probabilité non nulle d'être la cause d'un problème détecté. Enfin la liste des composantes déclarées fautives (L(F)) initialement vide contient l'identification des composantes reconnues défaillantes non encore isolées pour traitement.
Le principe de diagnostic des "Break-Fault" est simple et découle de leur caractéristique principale à demeurer permanente jusqu'à leur réparation. Il s'agit de définir les conditions d'appartenance d'une composante C à la liste des composantes fautives L(F).

Principe de diagnostic des "Break-Fault"

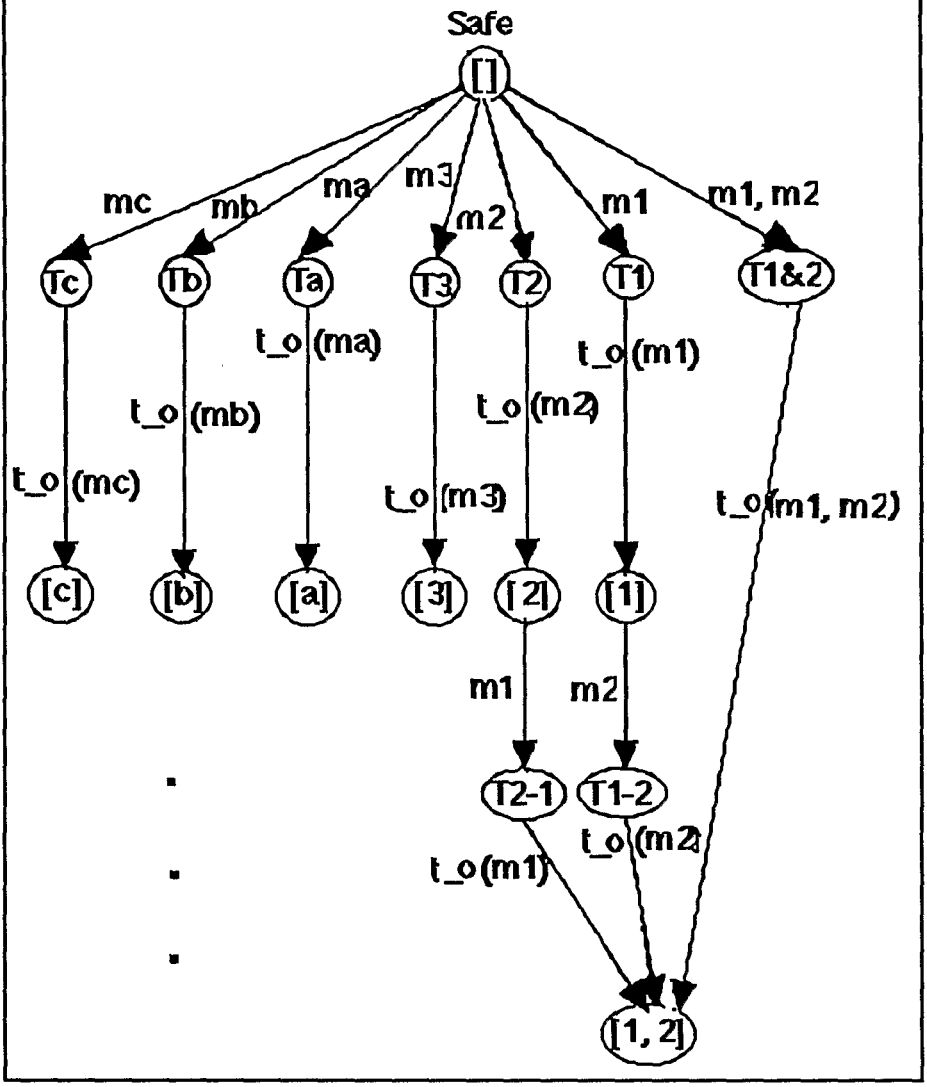
L(F) = { C/ pour tout ti < t < tf, S(t,C) = 0.0}
ti = Date d'apparition (constatée) de la présomption de faute
tf = ti+Delta (Delta = seuil de décision choisi judicieusement)
S(t,C) = Etat de santé de C à l'instant t

A une date d > tf, toute composante de L(F) est déclarée fautive ("down"). Le choix du seuil "Delta" doit être judicieusement fait afin d'éviter toute conclusion erronée.
Le diagnostic de plusieurs "Break-Fault" simultanées n'est pas simple. Il nécessite une solution plus élaborée. Nous utilisons par la suite les concepts de diagramme d'états finis de transition pour modéliser le comportement d'un domaine en présence de plusieurs fautes.

3.3.2 Comportement fautif du domaine

Le comportement du domaine peut être décrit par un diagramme d'états finis de transition qui sera mis à jour par le gestionnaire dès réception d'une nouvelle alarme ou d'une conclusion de diagnostic (nouveaux résultats de test). Nous décrirons les états de transition sous forme de combinaison de 0 (L(F) vide), 1 (L(F) a un élément), ou plusieurs (L(F) a plus d'un élément) fautes d'éléments du domaine (figure 8). Le nombre maximal de combinaisons possibles est au moins égale au nombre de sous-ensembles du domaine D (noeuds et liens). Ce nombre est réduit par la relation d'équivalence qui lie un certain nombre d'états. Les classes d'équivalence notées [ ]c représentent un ensemble de comportements fautifs équivalents du point de vue du diagnostic. Prenons le cas des états [1,2] et [1,2,a]. Comme la ligne (a) lie les noeuds (1) et (2), toute défaillance simultanée de (1) et (2) conduit à laisser inutilisée la ligne (a) jusqu'à réparation de l'un des noeuds. Ce qui donne un comportement similaire à celui observé lors de la défaillance simultanée de (1), (2) et de (a). Cette restriction ne doit pas affecter les opérations de surveillance ni celle de restauration ou de reconfiguration. Le nombre d'états dépend aussi du nombre de composantes, du degré de propagation des fautes, et de la tolérance aux fautes du système.
Le domaine peut être dans état opérationnel satisfaisant (safe). Cet état sera noté [ ]. Il peut aussi être dans un état temporaire (signalisation d'anomalie). Un nouveau message m (notification) signalant un problème a été reçu par le gestionnaire. La notation m_k indique que le message m se rapporte à un problème lié possiblement à l'objet k. Si le message reçu est nouveau, l'état du domaine passera alors de [ ] à un état T_k. L'objet k devient suspect par rapport au problème signalé. Un compteur relatif à m_k est alors chargé. Si le message m_k peut être corrélié à des messages précédents, seules les mises à jours des poids associés sont effectuées. A la reconnaissance d'un motif de faute ou au dépassement d'un seuil par le compteur C_k, un message time-out(m_k) est alors envoyé au gestionnaire. L'état du domaine passera de T_k à l'état [k] . Ce qui représente une faute permanente au niveau du noeud k .

FIGURE 8 : Diagramme de transition décrivant le comportement fautif du domaine de la figure 7



Les fautes les plus complexes sont les fautes multiples. Plusieurs composantes du domaine peuvent être fautives simultanément ou les unes après les autres. Dans ce qui suit nous décrivons le calcul de ces états et les transitions qui leurs sont associées.

Quelques exemples de classe d'états représentant des fautes multiples :

- [1,2]c = {[1,2], [1,2,a]} (1,2 fautives);
- [1,3]c = {[1,3], [1,3,b]} (1,3 fautives);
- [2,3]c = {[2,3], [2,3,c]} (2,3 fautives);
- [1,2,3]c = {[1,2,3], [1,2,3,a], [1,2,3,b], [1,2,3,c], [1,2,3,a,b], 1,2,3, a,c], [1,2,3,b,c], [1,2,3,a,b,c])} (1,2,3 fautives)

3.3.3 Calculs des états et des transitions

Le gestionnaire doit tout d'abord déterminer tous les états fautifs (si possible) de son domaine ainsi que les conditions de passage d'un état à un autre. A la réception d'une nouvelle notification, le gestionnaire détermine quelle transition effectuer à partir de l'état actuel du réseau. En la présence de plusieurs possibilités, la transition menant au diagnostic de la composante la plus prioritaire est exécutée. Un tirage au hasard sera effectué si les composantes concernées ont des priorités identiques. Un mécanisme de chaînage arrière (backtracking) doit être défini pour tenir compte des transitions écartées. Ces détails relèveront de nos futurs travaux.

Forme générale et exemple d'une transition menant à une faute :

[ ] ==> m_k ==> T_k ==> TimeOut(m_k)==> [k]

[ ] ====> m_1 ====>T_1====> TimeOut (m_1) ===> [1]

De l'état [ ] le gestionnaire peut passer à l'état T1 (transitoire) après réception d'un message (alerte) portant sur une faute possible du noeud 1. Un compteur C1 est chargé. A l'approche d'un seuil, un parcourt de l'arbre de la hiérarchie des composantes peut être lancé afin de détecter un motif de faute du noeud 1. Au dépassement du seuil relatif à C1 ou à la reconnaissance d'un motif de faute, une alarme est générée indiquant que la composante 1 est restée en permanence dans cet état anormal depuis un intervalle de temps significatif. La transition menant à l'état [1] est alors tirée.

Forme générale et exemple d'une transition menant à plusieurs fautes:

[ ] ==> m_k (k=1,..,n) ==> T_k ==> TimeOut(mx) ==> U[k] =[1,2,...,k]

[ ] ==>m_1,m_2==>T_1,T_2==> TimeOut (m_1,m_2) ===> [1,2]

De manière analogue à l'exemple précèdent, le gestionnaire peut passer de l'état [ ] à l'état T_1,T_2 après réception de messages (alerte) simultanées ou consécutives portant sur des fautes possibles des noeuds 1 et 3. Deux compteurs C1 et C2 sont chargés. A la reconnaissance des motifs des fautes ou aux dépassements des seuils relatifs à ces compteurs, une ou plusieurs alarmes sont générées indiquant que deux composantes sont restées en permanence dans cet

état anormal depuis un intervalle de temps significatif. La transition menant à l'état [1,2] est alors tirée. Si l'état courant est E au lieu de [ ], le nouvel état sera obtenu par l'union des états E et [1,2].

4. Conclusions

L'un des principaux avantages de l'approche de gestion proposée est la réduction de la complexité de la gestion de faute dans les réseaux. Dans cette approche de gestion, une analyse hiérarchique des alarmes est toujours effectuée au niveau local d'un domaine. Seules les conclusions finales portant sur la résolution d'un problème sont envoyées au niveau supérieur. Ce qui offre un cadre approprié de gestion de gros volumes d'alarmes. Des simulations graphiques ont été réalisées afin de valider l'approche de diagnostic proposée. Les opérations de configuration du réseau abstrait ont aussi été simulées. Pour l'instant les aspects temps réel des réseaux n'ont pas été considérés. Les travaux futurs seront orientés vers la vérification des solutions obtenues sur un réseau réel construit par l'interconnexion des réseaux locaux de l'Université de Montréal, du CRIM (Centre de Recherche Informatique de Montréal) et éventuellement de l'Université de Sherbrooke.

Remerciements :

Ce travail a été partiellement financé par le projet IGLOO (Ingénierie du Génie Logiciel Orienté Objet) et s'inscrit dans le cadre de l'axe application de ce projet. Nous tenons a présenter nos remerciements à toutes les personnes qui ont donné des remarques constructives sur nos travaux. Nous remercions en particulier le professeur Gregor v. Bochmann pour la lecture de cet article et les multiples suggestions faites.

5. Références

1. Sutter, M. T., *Designing Expert Systems for Real-Time Diagnosis of Self-Correcting Networks*, IEE NETWORK, page 43-51, September 1988.
2. Gaïti, D., *Un environnement réparti pour la gestion des réseaux*, Réseaux et informatique répartie, page 49-62, Volume 2-no 1/1992.
3. Guiagoussou, M., Anindya, D., Gregor, v. B., *An Overview of Fault Management in Telecommunication Networks*, Advanced Information Processing Techniques for LAN and MAN Management (C-17)/ J.-P. Claudé (Editor), Elservier Science Publishers B.V. (North Holland), page 69-85, April 1993.
4. Osborn, B. M., Withney, C. T., *Object Oriented Correlation*, BT Technol. Journal, Vol 11 No 3, page 43-51, 3 July 1993.