

12

DEMANDE DE BREVET D'INVENTION

A1

22 Date de dépôt : 04.11.13.

30 Priorité :

43 Date de mise à la disposition du public de la demande : 08.05.15 Bulletin 15/19.

56 Liste des documents cités dans le rapport de recherche préliminaire : *Se reporter à la fin du présent fascicule*

60 Références à d'autres documents nationaux apparentés :

○ Demande(s) d'extension :

71 Demandeur(s) : INSTITUT MINES-TELECOM / TELECOM SUDPARIS Etablissement public — FR.

72 Inventeur(s) : MAAG STEPHANE et CHE XIAOPING.

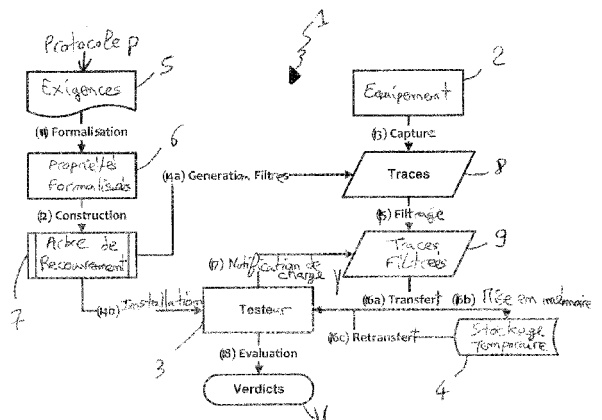
73 Titulaire(s) : INSTITUT MINES-TELECOM / TELECOM SUDPARIS Etablissement public.

74 Mandataire(s) : CABINET NONY.

54 ARCHITECTURE DE TEST DE PROTOCOLES.

57 La présente invention concerne un système de test (1) d'un protocole (P), notamment un protocole d'un réseau de communication, actif sur un équipement (2), notamment l'unité centrale d'un ordinateur, le système de test (1) comportant un moniteur configuré pour capturer de façon continue des traces (8), correspondant à des séquences de messages du protocole sous test (P), un filtre des traces capturées utilisant des propriétés formalisées des exigences du protocole sous test (P), notamment configuré pour séparer lesdites traces en données et en parties de contrôle, une zone de stockage temporaire (4), et un module de transfert agencé pour transférer les traces filtrées (9) à un testeur (3) configuré pour émettre des verdicts (V) en vérifiant si les traces satisfont les exigences des propriétés formalisées du protocole sous test (P),

système de test dans lequel, en cas de saturation du testeur (3) conduisant à l'émission d'une notification de charge (Y), le module de transfert interrompt le transfert des traces filtrées (9) vers le testeur (3) et met en mémoire au moins partiellement lesdites traces dans la zone de stockage temporaire (4), notamment uniquement leurs parties de contrôle, avant de reprendre le transfert des traces filtrées (9) lorsque le testeur (3) est à nouveau disponible.



La présente invention concerne un système de test dans le cadre de l'ingénierie des protocoles et services de communication, en particulier dans les réseaux, fournissant des verdicts pour des propriétés protocolaires fonctionnelles.

Les communications sont essentielles de nos jours et, un nombre de plus en plus important de services étant disponible en ligne, les réseaux d'ordinateurs continuent de grandir, et de nouveaux protocoles et services de communication sont constamment développés. Des standards de communication sont alors indispensables pour permettre aux différents systèmes d'interagir et de travailler ensemble. Ces standards sont formellement vérifiés, mais leurs implémentations peuvent contenir des fautes et doivent être testées, afin de vérifier si elles possèdent un ensemble de propriétés désirées et de comportements attendus.

Un protocole est un ensemble de règles de communication et de formats de messages permettant la communication entre des entités d'un système, distribué ou non.

Il existe deux approches pour tester la conformité et la performance d'un protocole de communication : un test dit actif, qui contrôle le protocole sous test et observe ses comportements par stimulation, et un test dit passif, qui collecte des traces d'exécution, notamment binaires, et les analyse a posteriori. Ces traces correspondent à des séquences de messages entrants et sortants du protocole, et comportent des parties de contrôle et des données, correspondant respectivement à l'entête et au corps des messages. Les traces sont généralement découpées en paquets de différentes tailles.

Le test passif est ainsi appliqué en « *post-capture* » : des traces finies, c'est-à-dire de taille déterminée, sont collectées et analysées. Cette approche est très difficilement automatisable et nécessite, dans de nombreux cas, un modèle formel du protocole, par exemple de type Unified Modeling Language (UML) ou Systems Modeling Language (SysML). En outre, travailler sur des traces finies augmente le nombre de verdicts inconclusifs, c'est-à-dire où le statut de la propriété sous test est incertain, et de faux-positifs.

Les solutions de test passif connues sont nombreuses. Elles sont proposées notamment par des opérateurs, des fournisseurs de services ou des laboratoires de recherche.

Certaines de ces approches sont liées à une spécification formelle du protocole sous test ou à un moniteur formel. C'est notamment le cas de travaux de monitoring en

ligne, appelé « *runtime monitoring* » en anglais, c'est-à-dire effectué pendant l'exécution du protocole sous test, développés dans des centres de recherche tels que Technische Universitat Munchen. L'inconvénient de ces approches est la nécessité de construire au préalable un modèle formel ou semi-formel du protocole sous test.

5 La demande WO 03/047280 décrit l'évaluation des performances d'un équipement GPRS ou 3G. Dans cette approche, c'est l'équipement qui est testé et non le protocole de communication, le cahier des charges du protocole n'étant ainsi pas formalisé.

 La demande US 2007/0124456 décrit une méthode de test hors ligne et nécessitant la collecte de traces. Les verdicts inconclusifs ne sont pas traités.

10 D'autres approches sont basées sur des systèmes de diagnostic et/ou de suivi. C'est notamment le cas d'outils fournis par des industriels, tels que par exemple les outils d'analyses d'EXFO ou de Microsoft Open Specification NM tool. Ces outils performants visent à contrôler et analyser les protocoles de communication, mais pas à les tester de manière passive contre des propriétés issues du cahier des charges et/ou des standards de
15 ces protocoles. Le test se fait ainsi en post-analyse, quand un opérateur alerte sur une observation inattendue ou un suivi anormal. Ces outils génèrent uniquement une vue du protocole dans le temps mais ne le testent pas passivement. Certains de ces outils fournissent des suites de test, ce qui est à rapprocher du test actif des protocoles.

 De nombreuses approches ne considèrent que la partie de contrôle des traces
20 protocolaires observées. La présence de verdicts faussement positifs est ainsi augmentée.

 En outre, certaines approches de test passif en ligne existent, ou « *runtime* » en anglais, comme les solutions proposées par l'entreprise Elvior, mais sont limitées par les tailles des mémoires tampon utilisées. Des coupures dans le test surviennent donc le temps de vider ces mémoires, par exemple pendant dix secondes, et peuvent conduire à un
25 nombre important de verdicts inconclusifs.

 Il existe un besoin pour tester passivement, en ligne, de manière efficace, non intrusive et continue, les implémentations des protocoles de communication en observant leurs traces d'exécution et en y appliquant certaines méthodes d'analyse permettant de fournir un verdict de test établissant si le protocole répond ou pas au cahier des charges
30 et/ou au standard.

L'invention a ainsi pour objet, selon un premier de ses aspects, un système de test d'un protocole, notamment un protocole d'un réseau de communication, actif sur un équipement, notamment l'unité centrale d'un ordinateur, le système de test comportant :

- 5 - un moniteur configuré pour capturer de façon continue des traces, correspondant à des séquences de messages du protocole sous test,
- un filtre des traces capturées utilisant des propriétés formalisées des exigences du protocole sous test, notamment configuré pour séparer lesdites traces en données et en parties de contrôle,
- une zone de stockage temporaire, et
- 10 - un module de transfert agencé pour transférer les traces filtrées à un testeur configuré pour émettre des verdicts en vérifiant si les traces satisfont les exigences des propriétés formalisées du protocole sous test,

système de test dans lequel, en cas de saturation du testeur conduisant à l'émission d'une notification de charge, le module de transfert interrompt le transfert des traces filtrées vers
15 le testeur et met en mémoire au moins partiellement lesdites traces dans la zone de stockage temporaire, notamment uniquement les parties de contrôle desdites traces, avant de reprendre le transfert des traces filtrées lorsque le testeur est à nouveau disponible.

L'invention fournit une architecture de test passif en ligne, non intrusive, transparente et produisant des verdicts de test de manière autonome en analysant
20 l'exécution d'un protocole en ligne dans son contexte, sans surcharger le processeur et la mémoire de l'équipement où fonctionne le système de test et où est actif le protocole. Les utilisateurs du réseau peuvent continuer à travailler pendant le test. Moins il y a d'activité sur l'équipement, plus le système de test peut travailler.

Pour mettre en œuvre l'invention, il n'est pas nécessaire d'avoir un modèle
25 formel du protocole sous test, et les propriétés protocolaires issues du standard et/ou du cahier des charges du protocole sont uniquement formalisées.

L'analyse des traces est réalisée en ligne, sans collecte préalable, ce qui permet d'automatiser le test sur des traces de taille quasiment infinie et de réduire dans un même temps le nombre de faux-positifs et de verdicts inconclusifs.

30 L'observation du protocole peut avoir lieu en continu, sans jamais arrêter le protocole et sans jamais interrompre l'enregistrement des traces, malgré leur taille. La notification de charge permet notamment de ne pas perdre de données bien que le test soit

momentanément interrompu, afin d'alléger les filtres et les ressources utilisées sur l'équipement.

Formalisation de protocoles

Un message d'un protocole de communication est de préférence une collection de champs de données de différents domaines. Les domaines de données peuvent être définis comme atomiques, « *atomic* » en anglais, ou composés, « *compound* » en anglais. Le domaine atomique peut être défini comme un ensemble de valeurs numériques ou de chaînes de caractères.

Une valeur composée v de taille $n > 0$ est définie par un ensemble de paires

10 $\{(l_i, v_i) \mid l_i \in L \wedge v_i \in D_i \cup \{\epsilon\}, i = 1 \dots n\}$, où $L = \{l_1, \dots, l_n\}$ est un ensemble prédéfini de labels et D_i représentent les domaines de données. Le domaine composé peut alors être défini comme l'ensemble de toutes les valeurs avec un même ensemble de labels et de domaines défini par $\langle L, D_1, \dots, D_k \rangle$.

Pour un protocole réseau P donné, un domaine composé M_p peut généralement être défini par un ensemble de labels et de domaines de données dérivés du format du message défini dans le cahier des charges du protocole. Un message du protocole P est tout élément $m \in M_p$. Pour chaque $m \in M_p$, un nombre réel $t_m \in \mathbb{R}^+$ est ajouté, représentant le moment où le message est reçu ou envoyé par l'équipement sous test.

Par exemple, un message possible pour le protocole SIP (Session Initiation Protocol, portant la référence IETF RFC3261), spécifié en utilisant la définition précédente, est :

$$m = \{(method, 'INVITE'), (time, '644.294133000'), (status, \epsilon), (from, 'alice@a.org'), (to, 'bob@b.org'), (cseq, \{(num, 7), (method, 'INVITE')\})\}$$

Ce message représente une requête INVITE de alice@a.org à bob@b.org. La valeur de *time* '644.294133000' ($t_0 + 644.294133000$) est une valeur relative, l'interface du protocole, appelée PO, lançant un minuteur, correspondant à la valeur initiale t_0 , lorsqu'il capture des traces. L'interface fournit également l'intervalle de temps relatif $T \subset \mathbb{R}^+$ pour tous les messages m de chaque trace.

Une *trace* peut être définie comme une séquence de messages du même protocole contenant les interactions entre un équipement sous test dans un réseau, par le

biais d'une interface PO, avec un ou plusieurs pairs du réseau pendant une période de temps arbitraire. Les traces sont avantageusement découpées en paquets de données.

Un *terme* peut être défini dans la forme de Backus-Naur (BNF) par $term ::= c \mid x \mid x.l.l...l$, où c est une constante dans un certain domaine, x est une variable, l représente un label, et $x.l.l...l$ est appelé un *sélecteur de variable*.

Une *substitution* est un ensemble fini de liaisons $\theta = \{x_1/term_1, \dots, x_k/term_k\}$, où chaque $term_i$ est un *terme* et où x_i est une variable telle que $x_i \neq term_i$ et $x_i \neq x_j$ si $i \neq j$.

Un *atome* peut être défini par :

$$A ::= \overbrace{p(term, \dots, term)}^k \begin{array}{l} \mid term = term \\ \mid term \neq term \\ \mid term < term \\ \mid term + term = term \end{array}$$

10

où $p(term, \dots, term)$ est un prédicat de label p et d'arité k . L'*atome temporisé* est un

atome particulier défini par $\overbrace{p(term_t, \dots, term_t)}^k$, où $term_t \in T$.

Les relations entre termes et atomes peuvent être établies par des définitions de *clauses*. Une clause est une expression du type : $A_0 \leftarrow A_1 \wedge \dots \wedge A_n$, où A_0 est l'entête de la clause et $A_1 \wedge \dots \wedge A_n$ le corps de la clause, A_i étant des atomes.

15

Une formule est définie en BNF par l'expression suivante :

$$\phi ::= A_1 \wedge \dots \wedge A_n \mid \phi \rightarrow \phi \mid \forall_x \phi \mid \forall_{y>x} \phi \\ \mid \forall_{y<x} \phi \mid \exists_x \phi \mid \exists_{y>x} \phi \mid \exists_{y<x} \phi$$

où A_1, \dots, A_n sont des atomes, $n \geq 1$ et x, y des variables utilisées pour formellement spécifier le message d'une trace. Les quantificateurs \exists et \forall sont communément utilisés pour définir les termes mathématiques « *il existe* » et « *pour tout* », respectivement.

20

Ainsi, la formule $\forall_x \phi$ signifie « *pour tous les messages x dans la trace, ϕ est satisfait* ». En se fondant sur les opérateurs et quantificateurs précédemment décrits,

l'interprétation des formules en verdicts T « PASS », \perp « FAIL », ou ? « INC », peut être réalisée.

Système de test

5 La zone de stockage temporaire, le moniteur, le module de transfert et le testeur du système de test selon l'invention peuvent être intégrés dans l'équipement sous test. Dans une variante, l'ensemble de ces éléments ou certains de ces éléments sont externes à l'équipement.

Le testeur peut comporter une mémoire tampon.

10 La mise en mémoire des traces filtrées dans la zone de stockage temporaire peut être réalisée en utilisant des tables de hachage.

Cinq types distincts de verdicts de test par analyse fonctionnelle des observations des traces peuvent être fournis : « PASS », la trace satisfait les exigences, « FAIL », la trace ne satisfait pas les exigences, « TIME-FAIL », hors-délai, « INC », inconclusifs, et « temp-INC », temporairement inconclusifs.

15 Les verdicts hors-délai peuvent survenir du fait que le test se fait en ligne, et en temps réel. Un temps de repos est avantageusement utilisé pour arrêter de chercher le message ciblé dans le but de fournir des statuts en temps réel. Ce temps de repos est généralement le temps maximal de réponse écrit dans le protocole. Si le message ciblé ne peut être observé avant le temps de repos, un verdict « TIME-FAIL » sera assigné à cette
20 propriété. Ce type de verdict n'est fourni que s'il n'y a pas de contrainte de temps dans les exigences. Dans ce dernier cas, la violation de cette exigence se traduira par un verdict « FAIL ».

Les verdicts inconclusifs peuvent correspondre au cas où une requête a été envoyée et où on attend la réponse, le statut de la propriété étant incertain. Ces verdicts
25 surviennent principalement au début et à la fin d'un test. Si une propriété nécessite de tester les données et que, à cause du filtrage, il n'y a pas de données dans la trace mais uniquement des parties de contrôle, un verdict temporairement inconclusif peut être assigné à la propriété, et peut ensuite changer, en « PASS » ou en « FAIL », lorsque les données recommencent à arriver de la zone de stockage.

30 Les verdicts de test peuvent être stockés dans des fichiers, afin d'être consultés ultérieurement pour l'évaluation du protocole.

Test de protocoles

La formalisation des exigences informelles du protocole sous test peut être réalisée par le système de test. Cette formalisation peut être réalisée en utilisant une syntaxe et une sémantique fondées sur la logique de Horn, et peut conduire à l'établissement de formules, comme décrit dans les articles «*Data-centric property formulation for passive testing of communication protocols*» de Lalanne et al, IASME/WSEAS, ACC'11/MMACTEE'11, pages 176 à 181, et «*A formal data-centric approach for passive testing of communication protocols*», de Lalanne et al, IEEE/ACM Transactions on networking. La logique de Horn permet d'avoir une syntaxe simplifiée.

Dans une variante, la formalisation des exigences informelles du protocole sous test est réalisée par un module externe, en amont du procédé de test, les formules résultantes étant fournies au système de test, par exemple sous la forme de fichiers, tout en respectant la syntaxe de formalisation des exigences du protocole.

Un arbre de recouvrement indexé, ou «*index spanning tree*» en anglais, est ensuite avantageusement construit à partir de toutes les formules représentant les propriétés formalisées. Chaque quantificateur de chaque formule parcourt l'arbre pour construire les nœuds, ou «*nodes*» en anglais. L'avantage d'un tel arbre est de supprimer des liens redondants et de ne garder que les liens les plus intéressants.

L'arbre peut être construit en mettant en œuvre l'algorithme suivant :

<p>Algorithm Tree Generation(K, T, S)</p> <p>Input: Set of Quantifiers S, Quantifiers K. Quantifier K containing the clauses and atoms. T represents the final spanning tree</p> <p>Output: \top if the Tree T has a solution.</p> <p>(a) if S is not empty then</p> <p>(b) for $K \in S$ and K is not empty do</p> <p>(c) $A \leftarrow \text{pop}(K)$</p> <p>(d) $\text{solved} \leftarrow \perp$</p> <p>(e) for $(B_0 \leftarrow B_1 \wedge \dots \wedge B_q) \in K$ do</p> <p>(f) for $(0 < i < q, 0 < j < m)$ do</p> <p>(g) if $(B_i \in T_j)$ then</p> <p>(h) if $(\neg \text{exist}(B_i, T_j))$ then</p> <p>(i) build(B_i, T_j)</p> <p>(j) else $i++$</p> <p>(k) push($B_0, B_1 \wedge \dots \wedge B_q, K$)</p> <p>(l) pop($B_0, B_1 \wedge \dots \wedge B_q, K$)</p> <p>(m) else $j++$</p> <p>(n) $\text{solved} \leftarrow \top$</p> <p>(o) $K \leftarrow \text{pop}(S)$</p> <p>(p) else useSolution(S)</p> <p>(q) return \top</p>

L'arbre de recouvrement peut être construit par le système de test. Dans une variante, l'arbre de recouvrement est construit par un module externe, en amont du procédé de test, et est fourni au système de test, par exemple sous la forme de fichiers.

5 Le moniteur peut capturer de manière continue et consécutive les traces du protocole actif sur l'équipement sous test. Lorsque les messages sont capturés, chaque message est de préférence horodaté.

L'arbre de recouvrement est avantageusement appliqué aux traces capturées, jouant ainsi le rôle d'un filtre. Dans le même temps, l'arbre est également envoyé au testeur avec la définition des verdicts, calculés par la procédure décrite dans l'article de
10 Che et al, « *A logic-based passive testing approach for the validation of communicating protocols* », conférence ENASE 2012, Pologne, pages 53 à 64. L'arbre permet d'optimiser le calcul des verdicts tout en considérant les données pertinentes transportées.

Les traces capturées peuvent parcourir l'arbre, et être ensuite filtrées en différents ensembles de données. Les messages non nécessaires, c'est-à-dire non requis par
15 les exigences du protocole, peuvent être filtrés dans un ensemble « Inconnu », ou « *Unknown* » en anglais, et ne sont, de préférence, pas pris en compte dans la suite du test. Cette étape permet de réduire le temps de traitement, puisque des comparaisons inutiles concernant des messages non pertinents sont évitées.

Les traces filtrées peuvent être transférées au testeur quand ce dernier a les
20 capacités nécessaires pour poursuivre le test. Si la priorité du testeur doit être baissée, afin de décharger le processeur et/ou la mémoire de l'équipement sous test, par exemple car ils doivent être utilisés pour une autre tâche par l'utilisateur, une notification de charge est avantageusement émise dans le but d'arrêter le transfert des traces filtrées au testeur, et de les mettre en mémoire dans l'espace de stockage temporaire.

25 Dans une variante ou en combinaison, quand la mémoire tampon du testeur atteint la limite de quantité de données pouvant être traitées, une notification de charge peut également être émise pour interrompre le transfert des traces filtrées et les stocker dans la zone de stockage temporaire. Plus précisément, lorsque les traces filtrées sont transférées à la mémoire tampon du testeur, une fonction servant à vérifier si cette
30 mémoire est en dépassement, ou « *overflow* » en anglais, peut être utilisée. Si la mémoire tampon a déjà atteint sa capacité maximale, l'équipement peut être notifié afin de rediriger les traces vers la zone de stockage temporaire pour éviter le dépassement. Au contraire, si

la mémoire tampon du testeur est dans un état stable, une notification de disponibilité peut être envoyée à la zone de stockage temporaire, pour relâcher les traces stockées, et à l'équipement, pour retourner au processus normal de transfert des traces.

5 Les traces sont ainsi doublement filtrées, par l'arbre de recouvrement et par le principe de notification de charge.

Selon le format des messages du protocole sous test, différentes méthodes de mise en mémoire temporaire peuvent être appliquées. Si la taille de l'entête du message est plus grande que celle de son corps, c'est-à-dire s'il y a plus de parties de contrôle que de données, le message entier peut être mis en mémoire dans l'espace de stockage temporaire.

10 Au contraire, dans le cas où la taille de l'entête du message est inférieure ou égale à la taille de son corps, c'est-à-dire dans le cas où il y a moins de parties de contrôle que de données, seules les parties de contrôle sont mises en mémoire dans l'espace de stockage temporaire. Mettre en mémoire uniquement les parties de contrôle des traces permet de préserver la mémoire pour pouvoir mettre en mémoire plus de messages, toutes les

15 exigences d'un protocole n'ayant pas forcément spécifiquement besoin d'utiliser les données.

Quand le testeur est à nouveau disponible, une nouvelle notification de charge est de préférence émise pour relâcher les traces temporairement stockées, et pour poursuivre le transfert de nouvelles traces. Dans le cas où uniquement les parties de

20 contrôle des traces ont été stockées, et si les exigences du protocole ont besoin d'utiliser les données, alors un verdict temporairement inconclusif, « temp-INC », peut être rendu. Ce verdict peut éventuellement être modifié par l'analyse de données provenant de traces entières lorsque le testeur est à nouveau disponible. Le verdict temporairement inconclusif peut alors devenir un verdict PASS ou FAIL dans la suite du test.

25 Le testeur vérifie avantageusement si les traces reçues satisfont ou pas les exigences formalisées afin de fournir des verdicts finaux, en les comparant notamment au comportement attendu du protocole spécifié par les propriétés formalisées.

Certaines propriétés peuvent être évaluées à partir de traces contenant uniquement des parties de contrôle, ou éventuellement très peu de données.

30 L'étape d'évaluation des formules se fonde avantageusement sur l'algorithme de résolution SLD (Sélectionné, Linéaire, Défini).

Procédé

L'invention a encore pour objet, selon un autre de ses aspects, un procédé de test d'un protocole, notamment un protocole d'un réseau de communication, actif sur un équipement, notamment l'unité centrale d'un ordinateur, procédé dans lequel :

- 5 - un moniteur capture de façon continue des traces, correspondant à des séquences de messages du protocole sous test,
- lesdites traces capturées sont filtrées en utilisant des propriétés formalisées des exigences du protocole sous test, les traces filtrées étant notamment séparées en données et en parties de contrôle, et
- 10 - lesdites traces filtrées sont transférées à un testeur configuré pour émettre des verdicts en vérifiant si les traces satisfont les exigences des propriétés formalisées du protocole sous test,

procédé de test dans lequel, en cas de saturation du testeur, une notification de charge est émise afin d'interrompre le transfert des traces filtrées vers le testeur, lesdites traces sont
15 au moins partiellement mises en mémoire dans une zone de stockage temporaire, notamment uniquement leurs parties de contrôle, avant que le transfert des traces filtrées ne soit repris lorsque le testeur est à nouveau disponible.

Les caractéristiques énoncées ci-dessus pour le système de test s'appliquent au procédé de test.

Produit programme d'ordinateur

L'invention a encore pour objet, selon un autre de ses aspects, un produit programme d'ordinateur pour la mise en œuvre du procédé de test d'un protocole tel que décrit précédemment, le produit programme d'ordinateur comportant des instructions de manière à ce que :

- 25 - des traces, correspondant à des séquences de messages du protocole sous test, soient capturées de façon continue,
- lesdites traces capturées soient filtrées en utilisant des propriétés formalisées des exigences du protocole sous test, les traces filtrées étant notamment séparées en données et en parties de contrôle, et
- 30 - lesdites traces filtrées soient transférées à un testeur configuré pour émettre des verdicts en vérifiant si les traces satisfont les exigences des propriétés formalisées du protocole sous test,

produit programme d'ordinateur comportant des instructions de manière à ce que, en cas de saturation du testeur, une notification de charge soit émise afin d'interrompre le transfert des traces filtrées vers le testeur, lesdites traces soient au moins partiellement mises en mémoire dans une zone de stockage temporaire, notamment uniquement leurs parties de
5 contrôle, avant que le transfert des traces filtrées ne soit repris lorsque le testeur est à nouveau disponible.

Les caractéristiques énoncées ci-dessus pour le système de test s'appliquent au produit programme d'ordinateur.

10 Le produit programme d'ordinateur peut être réalisé directement sur le processeur de l'équipement sous test.

L'invention pourra être mieux comprise à la lecture de la description détaillée qui va suivre, d'exemples non limitatifs de mise en œuvre de celle-ci, et à l'examen du dessin annexé, sur lequel :

- 15 - la figure 1 est un schéma représentant l'architecture du système de test selon l'invention,
- la figure 2 est un schéma représentant la génération de l'arbre de recouvrement,
- la figure 3 représente des exemples de messages filtrés,
- les figures 4 et 5 représentent deux types de message,
- 20 - la figure 6 est un schéma représentant les étapes de mise en mémoire et de notification, et
- les figures 7 et 8 représentent des exemples de test d'un protocole selon l'invention.

25 L'architecture du système de test 1 d'un protocole P selon l'invention est représentée à la figure 1. Le système de test 1 est destiné au test d'un protocole P actif sur un équipement 2, le protocole P étant par exemple le protocole d'un réseau de communication, et l'équipement 2 l'unité centrale d'un ordinateur. Pour ce faire, le système de test 1 comporte un testeur 3 configuré pour émettre des verdicts V en vérifiant si les exigences des propriétés du protocole P sous test sont respectées.

30 Le système de test 1 est configuré pour formaliser, lors d'une étape 11, les exigences 5 des propriétés du protocole sous test P. Pour ce faire, dans l'exemple décrit, une syntaxe et une sémantique fondées sur la logique de Horn sont utilisées et conduisent à

un ensemble de formules représentant les propriétés formalisées 6. Un arbre de recouvrement 7 est ensuite construit, lors d'une étape 12, à partir de l'ensemble de formules 6 représentant les exigences formalisées des propriétés du protocole sous test.

5 Dans une variante, au moins une des étapes de formalisation des exigences 5 des propriétés du protocole sous test P et de construction de l'arbre de recouvrement 7 est réalisée par un ou plusieurs modules externes, en amont du procédé de test, les formules 6 et/ou l'arbre de recouvrement 7 étant fournis, le cas échéant, au système de test 1, par exemple sous forme de fichiers.

10 Un moniteur capture de façon continue, lors d'une étape 13, des traces 8 provenant de l'équipement 2, correspondant à des séquences de messages du protocole P sous test.

15 L'arbre de recouvrement 7 sert, dans l'exemple décrit, lors d'une étape 14a, à générer un filtre pour filtrer les traces capturées 8, lors d'une étape 15, utilisant ainsi les propriétés formalisées du protocole sous test P. Le filtre est avantageusement configuré pour séparer les traces 8 en données et en parties de contrôle, et pour filtrer les messages non requis par les exigences, qui ne seront pas pris en compte par le testeur 3 pour l'évaluation du protocole P.

20 Lors d'une étape 14b d'installation, l'arbre de recouvrement est en outre envoyé au testeur 3 afin de définir les verdicts V. Cette étape 14b prépare toutes les données de la propriété sous test.

Un module de transfert transfère, lors d'une étape 16a, les traces filtrées 9 au testeur 3, afin qu'il émette des verdicts V en vérifiant si lesdites traces 9 satisfont les exigences des propriétés formalisées du protocole P sous test, lors d'une étape d'évaluation 18.

25 En cas de saturation du testeur 3, une notification de charge Y est émise, le transfert des traces filtrées vers le testeur 3 est interrompu et ces dernières sont mises en mémoire, au moins partiellement, lors d'une étape 16b, dans une zone de stockage temporaire 4 du système de test 1.

30 Le testeur 3 sature par exemple lorsque sa priorité baisse, pour décharger l'équipement 2 sous test.

Dans une variante ou en combinaison, le testeur 3 sature lorsqu'il atteint une limite de quantité de traces 9 pouvant être traitées.

Lorsque le testeur 3 est à nouveau disponible, une nouvelle notification de charge est de préférence émise pour relâcher, lors d'une étape 16c, les traces 9 temporairement mises en mémoire dans la zone de stockage 4, et pour poursuivre le transfert de nouvelles traces.

5 Les traces recueillies lors de l'étape de capture 13 sont, dans l'exemple décrit, des traces brutes issues du moniteur, appelé « *sniffer* » en anglais. Le filtrage de l'étape 15 permet d'enregistrer le flux entrant et sortant sur le protocole sous test P.

La mise en mémoire des traces 9 dans la zone de stockage temporaire 4 est avantageusement réalisée en utilisant des tables de hachage.

10 Le testeur 3 est configuré pour émettre cinq types de verdicts V, « PASS », « FAIL », « TIME-FAIL », « INC » et « temp-INC ». Les verdicts V sont de préférence stockés dans des fichiers.

Préférentiellement, toutes les étapes précédemment décrites sont accomplies, et, encore plus préférentiellement, elles sont accomplies dans l'ordre décrit.

15 La figure 2 représente la procédure de construction de l'arbre de recouvrement 7 à partir d'une formule 6. Chaque quantificateur Q de la formule 6 parcourt l'arbre 7 pour construire les nœuds A.

La figure 3 montre un exemple de messages du protocole SIP filtrés par l'arbre de recouvrement 7. D'après le standard SIP, une requête, dans le corps du message, est définie par le champ « *method* » et une réponse par le champ « *status* ». Les messages
20 peuvent donc avantageusement être filtrés et stockés selon l'arbre de recouvrement 7. Si un message n'est pas reconnu (notification « *inconnu* »), il sera non retenu grâce au filtrage.

La figure 4 représente un exemple de message où la taille de l'entête est supérieure à celle du corps.

25 La figure 5 représente un exemple de message où la taille de l'entête est inférieure ou égale à celle du corps. Dans ce cas, et en cas de saturation du testeur 3, seules les parties de contrôle des traces filtrées correspondant à ce type de messages sont mises en mémoire dans la zone de stockage temporaire 4 lors de l'étape 16b.

La figure 6 représente le processus de transfert et/ou de mise en mémoire des
30 traces filtrées 9 selon la charge du testeur 3. Le testeur 3 comporte une mémoire tampon 31 et une fonction vérifiant si ladite mémoire tampon 31 est en dépassement lorsque des traces filtrées 9 lui sont transférées. Si la mémoire tampon 31 du testeur 3 a déjà atteint sa

capacité maximale, l'équipement 2 reçoit une notification de charge Y afin de rediriger les traces 9 vers la zone de stockage temporaire 4 pour éviter le dépassement. Au contraire, si la mémoire tampon 31 du testeur 3 est dans un état stable, une notification de disponibilité N est envoyée à la zone de stockage temporaire 4 pour relâcher les traces stockées et à
5 l'équipement 2 pour retourner au processus normal de transport des traces.

La figure 7 représente un exemple de test selon l'invention. L'exigence 100 du protocole P dont la performance doit être testée est « Toute réponse 2xx pour une requête INVITE doit être répondue avec un ACK en 2 secondes ». Cette exigence est formalisée à l'étape 111 par la formule 101. L'arbre de recouvrement 102 correspondant à cette formule
10 101 est généré lors d'une étape 112. L'arbre 102 est transféré au processeur de l'équipement 2 pour être appliqué comme filtre, aux étapes 113 et 114, aux messages capturés du protocole P.

Dès que le processeur de l'équipement 2 a capturé une trace de 60 messages, filtrée à l'étape 115, il va vérifier la valeur de la notification de charge. Dans l'exemple
15 décrit, la valeur de notification de charge est égale à N, ce qui signifie que le testeur 3 est disponible et qu'il n'est pas nécessaire de mettre temporairement en mémoire les traces. Le processeur envoie directement les traces filtrées au testeur 3 à l'étape 116.

Lorsque le testeur 3 reçoit la trace, il la teste pour la propriété formalisée lors d'une étape 118, et renvoie la valeur de notification de charge N au processeur, lors de
20 l'étape 117, puisque le testeur est toujours disponible pour tester les traces suivantes. Sept verdicts « PASS » sont reçus dans l'exemple considéré, pour zéro verdicts « FAIL ».

Un autre exemple de test selon l'invention est représenté à la figure 8. La même exigence sous test 100 est utilisée et formalisée de la même manière en la formule 101, conduisant à l'arbre de recouvrement 102. 800 messages sont capturés et filtrés aux
25 étapes 123 et 124. Le processeur de l'équipement 2 vérifie la notification de charge à l'étape 125. Dans l'exemple considéré, le testeur est dans un état de priorité basse ce qui indique qu'il ne peut gérer un nombre important de traces à cet instant, une valeur Y de notification de charge est alors émise afin de mettre en mémoire les 600 traces entrantes en surplus dans la zone de stockage temporaire 106 dans une étape 126. Lorsque le testeur 3 a
30 fini de tester les 200 premières traces, il renvoie une valeur N de notification de charge au processeur et à la zone de stockage 106, afin de renvoyer au testeur 3, lors d'une étape 127,

les 600 messages mis en mémoire pour qu'il puisse les tester. 189 verdicts « PASS » sont reçus dans l'exemple considéré, et 32 verdicts « FAIL ».

L'invention n'est pas limitée aux exemples qui viennent d'être décrits.

5 L'équipement sous test 2 peut également être un équipement contenant une pile protocolaire et un espace mémoire, tel qu'un smartphone, un système embarqué, etc....

Le système de test selon l'invention permet de réaliser un nombre beaucoup moins élevé de comparaisons que les méthodes connues pour obtenir le verdict final, par exemple 26 comparaisons au lieu de 89 pour l'exemple de la figure 7.

10 Le domaine d'application du système de test selon l'invention concerne particulièrement l'ingénierie des protocoles et des services de communication. Le système de test passif peut s'appliquer à des protocoles des couches OSI 3 à 7 (Open Systems Interconnection), ou aux protocoles SIP et OLSR (Optimized Link State Routing, portant la référence RFC3626).

15 Le système de test selon l'invention peut être utilisé dans des systèmes communicants intégrant des protocoles ou services pouvant faire l'objet d'un monitoring. Il est particulièrement avantageux de pouvoir tester ces protocoles de manière passive sans perturber le comportement du système communicant et tout en l'utilisant à d'autres fins, telles que l'intégration d'autres composants, des simulations, un développement tierce ou l'analyse des performances.

20 Le système selon l'invention peut être utilisé dans des outils d'analyse et de monitoring des réseaux.

Le système de test selon l'invention peut être utilisé pour des systèmes distribués ou des systèmes non distribués. Dans le cas de systèmes distribués, un protocole peut être utilisé pour synchroniser les différents équipements.

25 L'expression « comportant un » doit être comprise comme étant synonyme de « comportant au moins un », sauf si le contraire est spécifié.

REVENDICATIONS

1. Système de test (1) d'un protocole (P), notamment un protocole d'un réseau de communication, actif sur un équipement (2), notamment l'unité centrale d'un ordinateur, le système de test (1) comportant :
- un moniteur configuré pour capturer de façon continue des traces (8), correspondant à des séquences de messages du protocole sous test (P),
 - un filtre des traces capturées utilisant des propriétés formalisées des exigences du protocole sous test (P), notamment configuré pour séparer lesdites traces en données et en parties de contrôle,
 - une zone de stockage temporaire (4), et
 - un module de transfert agencé pour transférer les traces filtrées (9) à un testeur (3) configuré pour émettre des verdicts (V) en vérifiant si les traces satisfont les exigences des propriétés formalisées du protocole sous test (P),
- 15 système de test dans lequel, en cas de saturation du testeur (3) conduisant à l'émission d'une notification de charge (Y), le module de transfert interrompt le transfert des traces filtrées (9) vers le testeur (3) et met en mémoire au moins partiellement lesdites traces dans la zone de stockage temporaire (4), notamment uniquement leurs parties de contrôle, avant de reprendre le transfert des traces filtrées (9) lorsque le testeur (3) est à nouveau
- 20 disponible.
2. Système selon la revendication 1, dans lequel un arbre de recouvrement indexé (7) est construit à partir des propriétés formalisées des exigences du protocole sous test (P).
3. Système selon la revendication précédente, dans lequel le filtre des traces capturées (8) est l'arbre de recouvrement indexé (7).
- 25
4. Système selon la revendication 2 ou 3, dans lequel l'arbre de recouvrement indexé (7) est en outre envoyé au testeur (3) afin de définir les verdicts (V).
5. Système selon l'une quelconque des revendications précédentes, dans lequel le filtre des traces capturées (8) est configuré pour filtrer les messages non requis
- 30 par les exigences, qui ne doivent pas être pris en compte par le testeur (3).

6. Système selon l'une quelconque des revendications précédentes, dans lequel le testeur (3) sature lorsque sa priorité est baissée, notamment pour décharger l'équipement sous test (2).

5 7. Système selon l'une quelconque des revendications précédentes, dans lequel le testeur (3) sature lorsqu'il atteint une limite de quantité de traces pouvant être traitées.

8. Système selon l'une quelconque des revendications précédentes, dans lequel le testeur (3) comporte une mémoire tampon (31) et une fonction vérifiant si ladite mémoire tampon est en dépassement.

10 9. Système selon la revendication précédente, dans lequel le testeur (3) envoie une notification de charge (Y) à l'équipement (2) si la mémoire tampon (31) du testeur (3) est en dépassement, afin de rediriger les traces filtrées (9) vers la zone de stockage temporaire (4).

15 10. Système selon l'une quelconque des revendications précédentes, dans lequel la taille de l'entête des messages du protocole (P) est inférieure ou égale à celle du corps des messages, et, en cas de saturation du testeur (3), seules les parties de contrôle des traces filtrées (9) correspondantes à ces messages sont mises en mémoire dans la zone de stockage temporaire (4).

20 11. Système selon l'une quelconque des revendications précédentes, dans lequel le testeur (3) est configuré pour émettre cinq types de verdicts (V), « PASS », « FAIL », « TIME-FAIL », « INC » et « temp-INC ».

12. Système selon l'une quelconque des revendications précédentes, dans lequel les verdicts de test (V) sont stockés dans des fichiers.

25 13. Système selon l'une quelconque des revendications précédentes, dans lequel les exigences des propriétés du protocole (P) sont formalisées en utilisant une syntaxe et une sémantique fondées sur la logique de Horn conduisant à un ensemble de formules (6), à partir desquelles un arbre de recouvrement indexé (7) est construit.

30 14. Procédé de test d'un protocole (P), notamment un protocole d'un réseau de communication, actif sur un équipement (2), notamment l'unité centrale d'un ordinateur, procédé dans lequel :

- un moniteur capture de façon continue des traces (8), correspondant à des séquences de messages du protocole sous test (P),

- lesdites traces capturées (8) sont filtrées en utilisant des propriétés formalisées des exigences du protocole sous test (P), les traces filtrées étant notamment séparées en données et en parties de contrôle, et
- lesdites traces filtrées (9) sont transférées à un testeur (3) configuré pour émettre des verdicts (V) en vérifiant si les traces satisfont les exigences des propriétés formalisées du protocole sous test (P),

procédé de test dans lequel, en cas de saturation du testeur (3), une notification de charge (Y) est émise afin d'interrompre le transfert des traces filtrées (9) vers le testeur (3), lesdites traces (9) sont au moins partiellement mises en mémoire dans une zone de stockage temporaire (4), notamment uniquement leurs parties de contrôle, avant que le transfert des traces filtrées (9) ne soit repris lorsque le testeur (3) est à nouveau disponible.

15. Produit programme d'ordinateur pour la mise en œuvre du procédé de test d'un protocole (P) tel que défini à la revendication précédente, le produit programme d'ordinateur comportant des instructions de manière à ce que :

- des traces (8), correspondant à des séquences de messages du protocole sous test (P), soient capturées de façon continue,
- lesdites traces capturées (8) soient filtrées en utilisant des propriétés formalisées des exigences du protocole sous test (P), les traces filtrées étant notamment séparées en données et en parties de contrôle, et

- lesdites traces filtrées (9) soient transférées à un testeur (3) configuré pour émettre des verdicts (V) en vérifiant si les traces satisfont les exigences des propriétés formalisées du protocole sous test (P),

produit programme d'ordinateur comportant des instructions de manière à ce que, en cas de saturation du testeur (3), une notification de charge (Y) soit émise afin d'interrompre le transfert des traces filtrées (9) vers le testeur (3), lesdites traces (9) soient au moins partiellement mises en mémoire dans une zone de stockage temporaire (4), notamment uniquement leurs parties de contrôle, avant que le transfert des traces filtrées (9) ne soit repris lorsque le testeur (3) est à nouveau disponible.

1/5

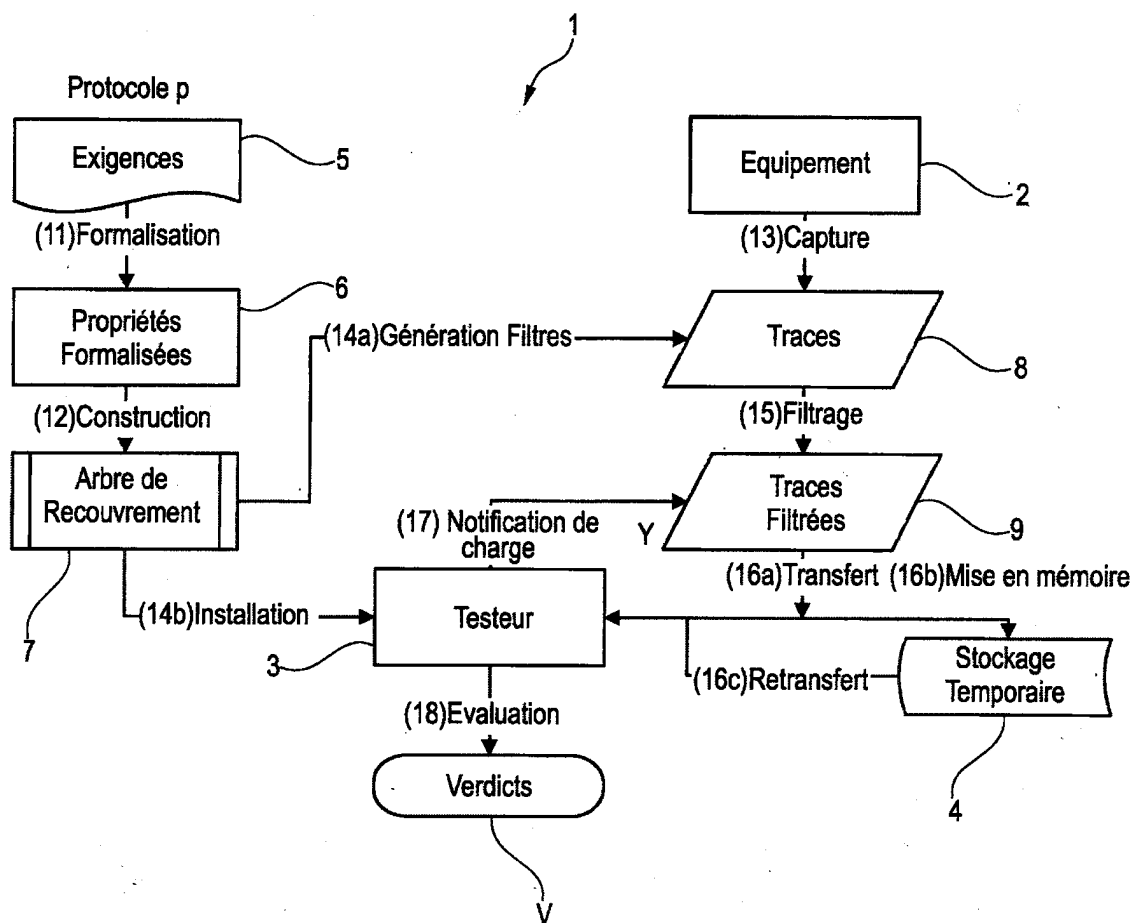


Fig. 1

2/5

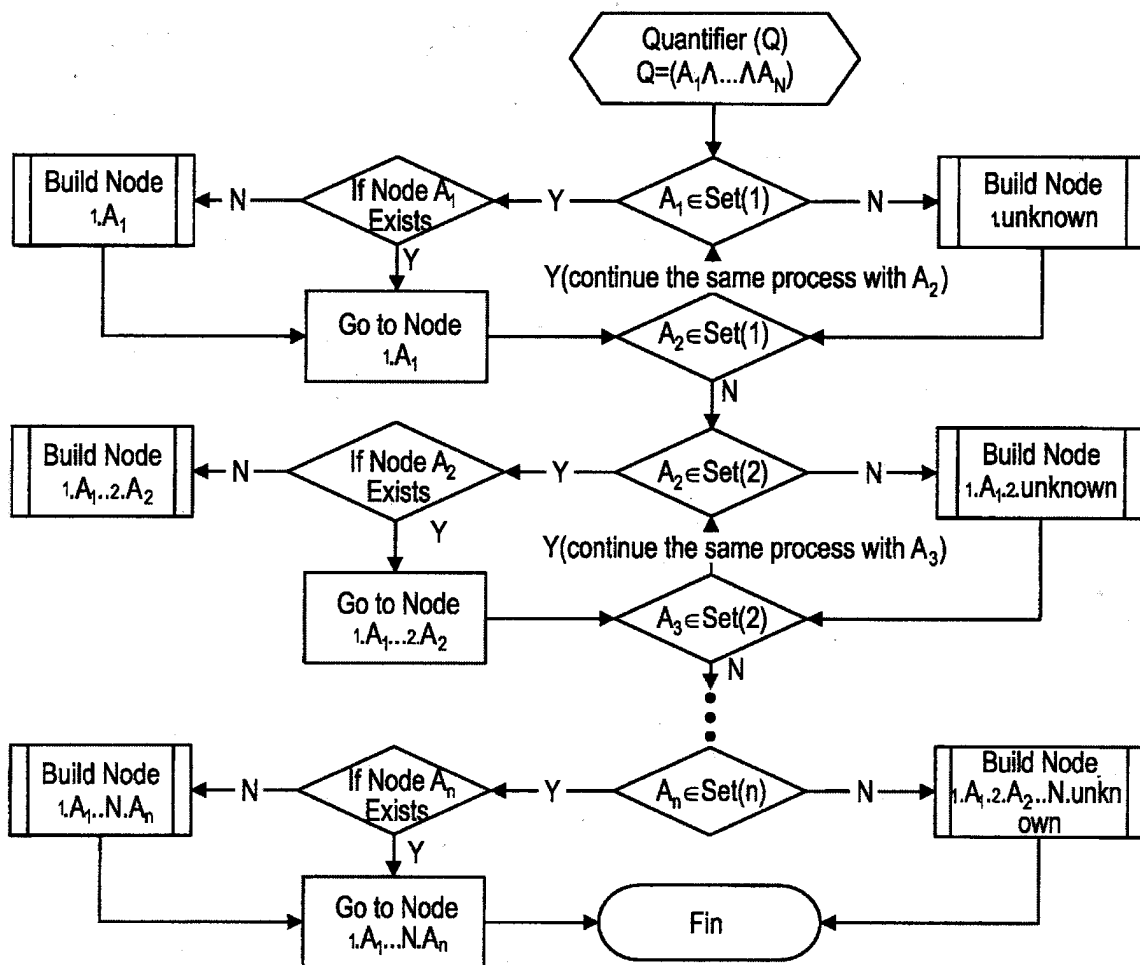


Fig. 2

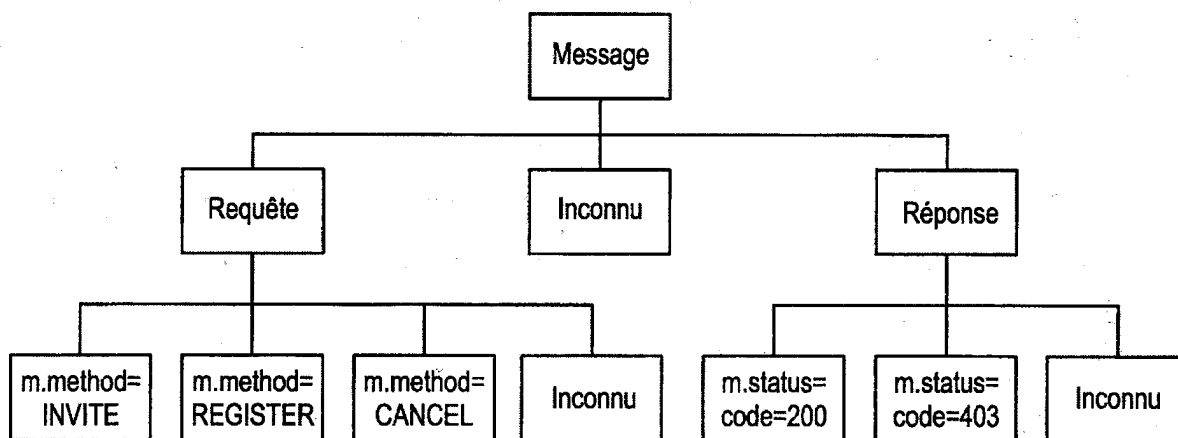


Fig. 3

3/5

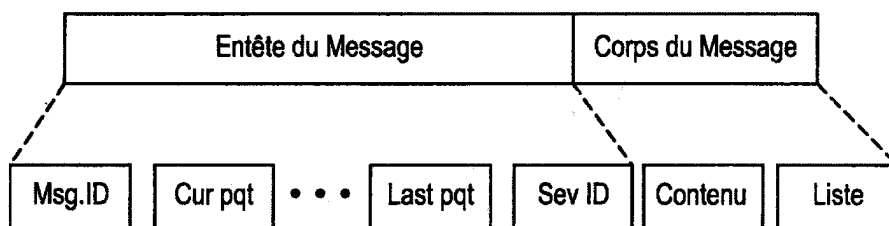


Fig. 4

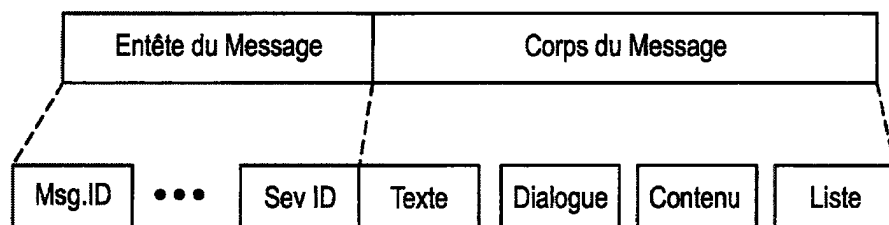


Fig. 5

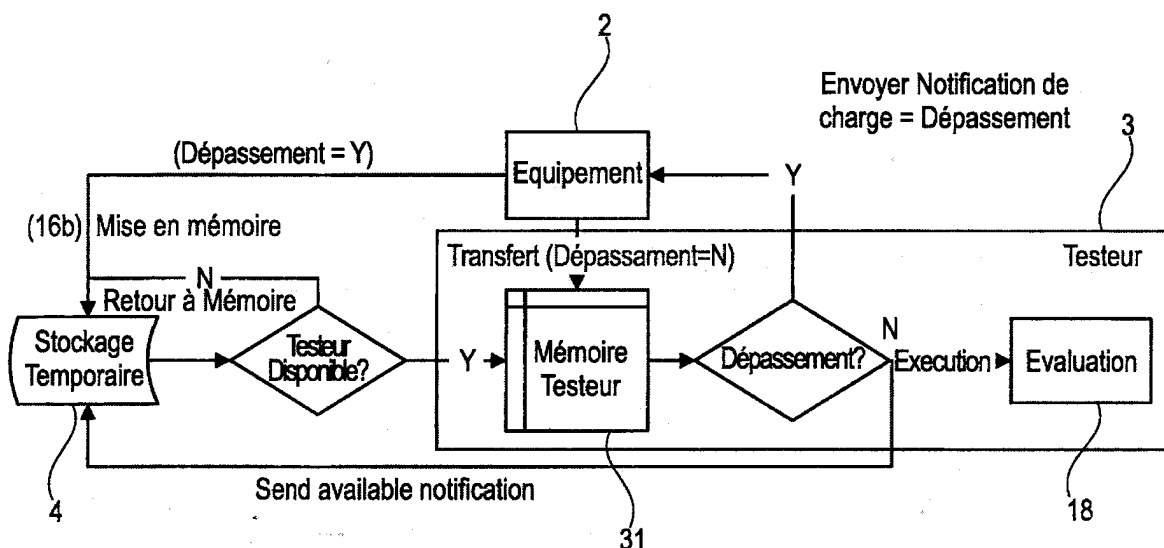


Fig. 6

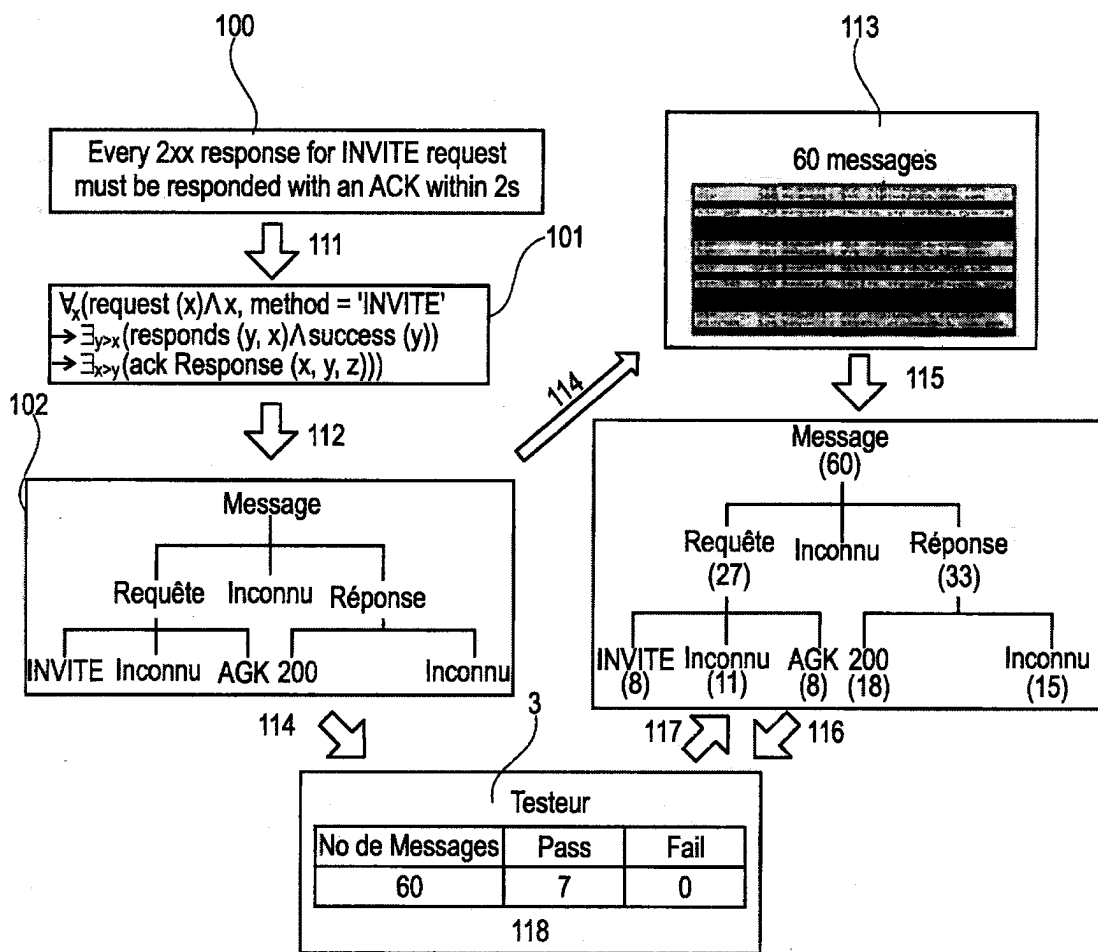


Fig. 7

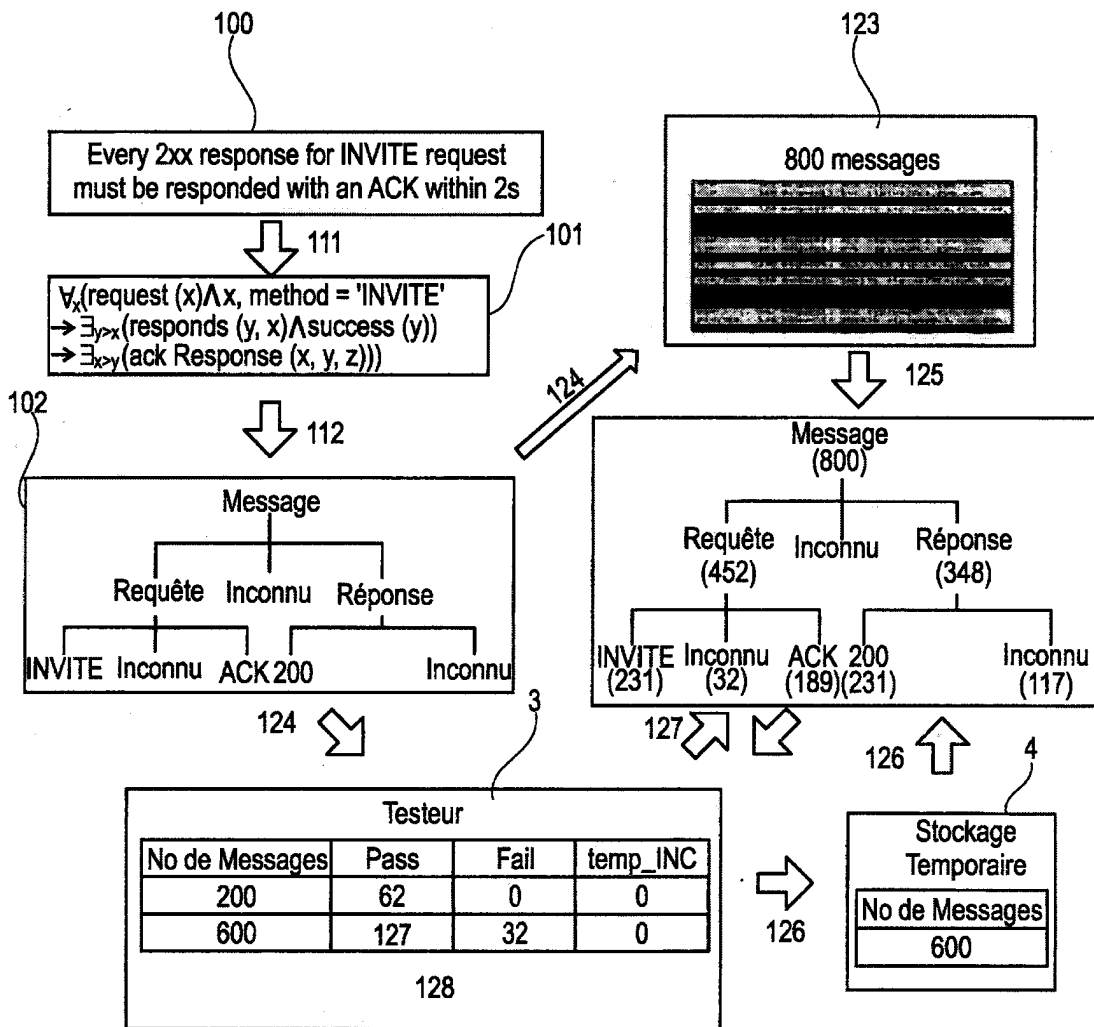


Fig. 8



**RAPPORT DE RECHERCHE
PRÉLIMINAIRE**

N° d'enregistrement national

établi sur la base des dernières revendications déposées avant le commencement de la recherche

FA 790929
FR 1360794

DOCUMENTS CONSIDÉRÉS COMME PERTINENTS		Revendication(s) concernée(s)	Classement attribué à l'invention par l'INPI
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes		
X	US 2004/054776 A1 (KLOTZ STEVE [US] ET AL) 18 mars 2004 (2004-03-18)	1,14,15	H04L29/08
A	* abrégé * * alinéas [0002], [0007] - [0012], [0029] - [0040], [0043] - [0045], [0065] - [0067], [0072] - [0074], [0082] - [0088] * * alinéas [0094], [0103] - [0106], [0115], [0170] - [0197] *	2-13	
A	US 2012/079324 A1 (APHALE ABHIJIT SUHAS [IN]) 29 mars 2012 (2012-03-29) * abrégé * * alinéas [0007] - [0010], [0025] - [0033], [0040] - [0043], [0053] - [0056] *	1-15	
A	Malik GHALLAB, Amine MOUNIR ALOUI: "Managing Efficiently Temporal RelationsThrough Indexed Spanning Trees", International Joint Conferences on Artificial Intelligence 25 août 1989 (1989-08-25), XP002726335, Chicago Extrait de l'Internet: URL:http://ijcai.org/Past%20Proceedings/IJ CAI-89-VOL-2/PDF/072.pdf [extrait le 2014-06-27] * abrégé *	2,3	DOMAINES TECHNIQUES RECHERCHÉS (IPC) H04L G06F
Date d'achèvement de la recherche		Examineur	
27 juin 2014		Le Bras, Patrick	
CATÉGORIE DES DOCUMENTS CITÉS		T : théorie ou principe à la base de l'invention	
X : particulièrement pertinent à lui seul		E : document de brevet bénéficiant d'une date antérieure	
Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie		à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure.	
A : arrière-plan technologique		D : cité dans la demande	
O : divulgation non-écrite		L : cité pour d'autres raisons	
P : document intercalaire		& : membre de la même famille, document correspondant	

1

EPO FORM 1503 12.99 (P04C14)

**ANNEXE AU RAPPORT DE RECHERCHE PRÉLIMINAIRE
RELATIF A LA DEMANDE DE BREVET FRANÇAIS NO. FR 1360794 FA 790929**

La présente annexe indique les membres de la famille de brevets relatifs aux documents brevets cités dans le rapport de recherche préliminaire visé ci-dessus.

Les dits membres sont contenus au fichier informatique de l'Office européen des brevets à la date du **27-06-2014**

Les renseignements fournis sont donnés à titre indicatif et n'engagent pas la responsabilité de l'Office européen des brevets, ni de l'Administration française

Document brevet cité au rapport de recherche	Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
US 2004054776 A1	18-03-2004	AUCUN	

US 2012079324 A1	29-03-2012	AUCUN	
