



NUMERO DE PUBLICATION : 1003811A3

NUMERO DE DEPOT : 8800365

MINISTERE DES AFFAIRES ECONOMIQUES

Classif. Internat.: H04L

Date de délivrance : 23 Juin 1992

Le Ministre des Affaires Economiques,

Vu la Convention de Paris du 20 Mars 1883 pour la Protection de la propriété industrielle;

Vu la loi du 28 Mars 1984 sur les brevets d' invention, notamment l' article 22;

Vu l' arrêté royal du 2 Décembre 1986 relatif à la demande, à la délivrance et au maintien en vigueur des brevets d' invention, notamment l' article 28;

Vu le procès verbal dressé le 30 Mars 1988 à 14h05
à l' Office de la Propriété Industrielle

ARRETE:

ARTICLE 1.- Il est délivré à : INDUSTRIAL TECHNOLOGY INSTITUTE
Hubbard 2901, Ann Arbor, MICHIGAN 48106(ETATS-UNIS D'AMERIQUE)

représenté(e)s par : GRISAR Daniel, OFFICE KIRKPATRICK, Square de Meeus, 4 - B
1040 BRUXELLES.

un brevet d' invention d' une durée de 20 ans, sous réserve du paiement des taxes annuelles, pour : SYSTEME DE TEST INTEGRE POUR TEST DE CONFORMITE DE SYSTEMES DE COMMUNICATION.

Priorité(s) 30.03.87 US USA 032185

ARTICLE 2.- Ce brevet est délivré sans examen préalable de la brevetabilité de l' invention, sans garantie du mérite de l' invention ou de l' exactitude de la description de celle-ci et aux risques et périls du(des) demandeur(s).

Bruxelles, le 23 Juin 1992
PAR DELEGATION SPECIALE :

WUYTS L.
Directeur.

Système de test intégré pour test de conformité de systèmes de communication.

Background et résumé de l'invention.

La présente invention concerne d'une manière générale un procédé pour tester le logiciel d'un système de communication. D'une manière plus spécifique, l'invention concerne un procédé pour déterminer si un système de communication est conforme à une spécification de protocoles de communication telle que la spécification MAP basée sur le modèle de système ouvert établi par l'Organisme International de Normalisation (ISO).

De nombreux procédés de fabrication sont actuellement régis par des systèmes informatiques interconnectés qui doivent communiquer entre eux. Dans l'intérêt de la normalisation, l'industrie automobile et d'autres industries de transformation ont proposé une spécification de protocoles de communication utilisés pour interconnecter des systèmes informatiques et des systèmes robotiques de telle sorte qu'ils communiquent entre eux. Une spécification de ce genre est la spécification MAP (Manufacturing Automation Protocol). La spécification MAP est basée sur un modèle de communication informatique établi par l'ISO et appelé "Modèle de système ouvert". Le modèle de système ouvert (et la spécification MAP) divisent le problème des communications en sept jeux de sorties apparentées, appelées couches, qui sont décrites plus loin.

Pour assurer une compatibilité avec le matériel informatique et le matériel d'automation existants d'un fabricant, les fournisseurs et les vendeurs de matériels nouveaux doivent élaborer un logiciel de communication compatible. Le matériel nouveau doit être conforme à la spécification MAP.

La spécification MAP et le modèle de système ouvert sur lequel elle est basée sont des protocoles de communication relativement compliqués. Tester un nouveau programme de communication ou un nouveau matériel de communication pour déterminer s'il est conforme à la spécification MAP n'est pas une mince affaire. Comme expliqué plus en détail plus loin, pour tester la conformité de toutes les couches d'un système de communication, il fallait jusqu'à présent que le réalisateur du nouveau produit écrive des programmes de test spéciaux et les incorpore directement dans le logiciel de communication. Ceci augmente la complexité et le coût du logiciel et peut laisser le système de communication avec du bagage logiciel supplémentaire qui n'est pas nécessaire dès que la conformité a été établie. Plus gênant encore peut s'avérer le fait que des sous-programmes de test de ce genre ont jusqu'à présent été liés étroitement à une couche particulière dans la spécification MAP. Cette liaison étroite a rendu très difficile l'intégration de certaines fonctions entre des couches adjacentes, sans que les sous-programmes de test n'y fassent obstacle.

L'invention évite cette difficulté par une technique de manipulation de mécanismes de réglage de flux de couche supérieure afin de gérer la transmission et la réception de messages de la couche inférieure à des fins de test de conformité. L'invention permet également de tester le nouveau produit sans nécessiter l'inclusion de certaines des fonctions de test spéciales requises précédemment dans le logiciel de nouveaux produits. L'invention permet de procéder au test de conformité sans interférence entre des couches adjacentes du logiciel en cours de test. L'invention permet donc au logiciel de communication d'utiliser des couches adjacentes intégrées ou interdépendantes, par

exemple entre les couches de réseau et de transport afin d'améliorer le rendement du logiciel.

D'une manière plus spécifique, l'invention permet d'effectuer le test de conformité de protocole sur la couche de réseau de l'implémentation de protocole sous test (IST), c'est-à-dire l'entité de protocole interréseau dans le système sous test, sans modification ou adjonction de code de test à l'interface réseau/transport (entre les couches de réseau et de transport) du produit. L'invention gère indirectement le flux de messages entre l'interface réseau/transport du produit afin de déterminer si la couche de réseau de l'implantation sous test passe des données à la couche de transport et en reçoit de cette couche correctement. L'invention procède au test de protocole de couche de réseau d'une manière non envahissante. Elle régit le flux d'informations à travers la couche de transport, afin de régler ainsi indirectement l'interface de service de réseau par l'intermédiaire du transport, en prenant indirectement le contrôle des signaux de crédit et d'accusé de réception engendrés automatiquement et périodiquement par la couche de transport.

Pour mieux comprendre l'invention, ses buts et ses avantages, on se référera ci-après à la description suivante, donnée à titre d'exemple avec référence aux dessins annexés.

Brève description des dessins.

La Fig. 1 est un schéma synoptique illustrant le protocole de système de communication à sept couches;

la Fig. 2 est un schéma synoptique illustrant la topologie de communication physique mise en oeuvre par le système de communication préféré actuellement;

la Fig. 3 est un schéma synoptique illustrant

un système intermédiaire ou routeur pour transférer des messages entre des noeuds non adjacents;

la Fig. 4 est un diagramme de séquence de messages illustrant un flux de transmissions de messages et d'accusés de réception entre initiateur et destinataire;

la Fig. 5 est un diagramme illustrant le principe de la fenêtre utilisé par la couche de transport du système de communication pour régler le flux d'informations;

la Fig. 6 est un schéma synoptique du programme de test de l'invention, illustrant la manière selon laquelle un exemple de programme de système de communication est testé;

la Fig. 7 est un diagramme illustrant la forme des unités de données de protocole à chacun des sept niveaux, et

la Fig. 8 (8a et 8b) est un organigramme illustrant la méthode de test conforme à l'invention.

Description de la forme d'exécution préférée.

Avant de fournir une explication détaillée de l'invention, on exposera brièvement le détail de la spécification MAP. La spécification MAP est décrite dans, General Motors, "Manufacturing Automation Protocol Version 2.1", General Motors Corporation, 1985. La spécification MAP est basée sur le modèle d'interconnexion de systèmes ouverts (Open Systems Interconnection ou OSI) établi par l'ISO. Une explication détaillée du modèle OSI peut être trouvée dans ISO IS7498 "Information Systems Processing - Open Systems Interconnection - Basic Reference Model", 1984, qui est cité ici à titre de référence.

La Fig. 1 illustre schématiquement les sept couches représentant la subdivision du protocole de communication MAP. Le niveau inférieur ou niveau le

plus primitif est le premier niveau, la couche physique. La couche physique est responsable du codage et de la transmission physique de messages entre deux noeuds adjacents du réseau. La norme MAP utilise une topologie à bus illustrée sur la Fig. 2. La couche physique met en oeuvre la norme IEEE 802.4. La communication s'effectue par l'intermédiaire d'un câble coaxial 10 qui fournit deux canaux de communication. Un canal est un canal à basse fréquence à propagation dans un premier sens et l'autre canal est un canal à haute fréquence à propagation dans le sens opposé. Une unité de tête de réseau 12 termine une extrémité du câble pour commuter les communications à basse fréquence vers la haute fréquence en vue de changer ainsi le sens de la propagation. Les unités de communication 14 placent des communications sur le câble coaxial 10 à la basse fréquence à laquelle elles se propagent vers l'unité de tête de réseau 12 en vue d'être retransmises à la fréquence supérieure. Toutes les unités de communication reçoivent des données à la haute fréquence.

Comme le montre la Fig. 1, immédiatement au-dessus de la couche physique se trouve le niveau deux, la couche de liaison de données. La couche de liaison de données est responsable de l'agencement des données binaires traitées par la couche physique en octets (8 bits). Plusieurs octets sont groupés en paquets de données par la couche de liaison de données. La couche de liaison de données lie une adresse de liaison à chaque paquet, cette adresse identifiant l'unité de communication 14 ou le noeud destiné à recevoir le paquet de données. La couche de liaison de données n'est responsable que de l'adressage des données destinées à des dispositifs branchés physiquement sur le câble coaxial. Pour envoyer un message vers un noeud non adjacent (un noeud qui n'est pas connecté physi-

quement au câble coaxial), on utilise une unité de système intermédiaire ou un routeur.

Le troisième niveau, c'est-à-dire la couche de réseau, traite le routage de l'information par des systèmes intermédiaires ou routeurs. La couche de réseau établit le protocole pour transférer des messages depuis un noeud sur un câble vers un noeud sur un câble séparé physiquement. La Fig. 3 illustre un système intermédiaire ou routeur 16 qui interconnecte un premier système de communication 18, communiquant par un câble coaxial 10, à un second système de communication 20, communiquant par un câble en fibres optiques 22. Le système intermédiaire ou routeur 16 utilise deux couches physiques 24 et 26 qui sont compatibles respectivement avec les protocoles pour câble coaxial et pour fibres optiques. Les couches de liaison de données 28 et 30 réagissent respectivement aux couches physiques 24 et 26 et communiquent à leur tour par l'intermédiaire de la couche de réseau 32 du routeur. La couche de réseau 32 du routeur peut être considérée comme étant un pont entre les couches de réseau respectives des premier et second systèmes de communication 18 et 20. De cette façon, il est possible d'établir des communications entre deux noeuds non adjacents utilisant des protocoles physiques semblables ou différents.

En théorie, la couche de réseau peut déterminer le noeud de destinataire d'une communication donnée par une variété de systèmes différents. L'un de ces systèmes est un protocole à connexion, dans lequel un canal virtuel permanent est créé entre deux appareils. Lorsqu'on utilise un tel protocole, il n'est pas nécessaire de placer une signature sur le message à transporter ni de placer une quelconque information de routage sur le message, car le canal virtuel permanent

décrit automatiquement l'identité de l'initiateur et du destinataire. Un autre protocole est le protocole sans connexion. Le protocole orienté sans connexion exige qu'un message soit enfermé dans une adresse ou accompagné de celle-ci et une signature identifiant le destinataire souhaité du message et l'initiateur du message. Le protocole sans connexion, contrairement au protocole à connexion, ne garantit pas que des messages soient transportés dans un ordre particulier quelconque. La spécification MAP adopte actuellement le protocole sans connexion. Ce protocole sans connexion est habituellement connu sous le nom de protocole interréseau ou PI.

Etant donné que le protocole PI ne garantit pas que des messages soient reçus dans l'ordre dans lequel ils ont été émis, des moyens doivent être prévus pour assurer l'acheminement et la distribution des messages. Comme le montre à nouveau la Fig. 1, le quatrième niveau ou couche de transport fournit l'assurance d'un acheminement certain des messages. La spécification MAP utilise une technique d'accusé de réception positive dans laquelle le destinataire d'un message accuse réception en renvoyant un signal d'accusé de réception (ACK) à l'initiateur. Les détails de ce système d'accusé de réception sont expliqués ci-après.

La couche de transport assure également le réglage du flux de messages. Le réglage de flux est le processus utilisé pour assurer que le rythme des communications soit approprié pour les deux dispositifs de communication. Par exemple, si une communication doit être établie entre un gros ordinateur à grande vitesse et un ordinateur personnel à basse vitesse, certains moyens doivent être prévus pour établir une vitesse de communication de travail. Ceci résulte d'un

processus de négociation entre les deux unités de communication et spécifiquement entre les couches de transport de ces deux unités. Les deux systèmes de communication se mettent d'accord avant la communication sur la durée maximum du message (ou unité de données de protocole, UDP). A mesure que les échanges de messages progressent, chaque unité de communication peut émettre des paramètres de réglage pour régler la vitesse de la transmission des UDP.

Le mécanisme par lequel la vitesse de transmission est réglée est basé sur le numéro d'ordre de l'UDP. Un numéro d'identification d'ordre unique est attribué à chaque UDP, de sorte que les numéros d'ordre peuvent être utilisés pour placer les paquets dans l'ordre voulu au cas où ils seraient mal classés. Lorsque la communication est établie, chaque unité de communication envoie à l'autre le numéro d'ordre correspondant au nombre maximum de UDP qu'elle est prête à recevoir. Les échanges de données se font en duplex intégral, chaque côté pouvant envoyer des UDP à l'autre jusqu'au numéro d'ordre maximum qui lui a été fourni par l'autre.

A titre d'exemple simple, si les deux unités amorcent une communication et que l'unité A est prête à recevoir trois UDP, elle doit envoyer le numéro 3 à l'unité B. Ensuite, si l'unité A souhaite que trois UDP supplémentaires soient émises, elle doit porter le numéro d'ordre à 6. L'unité B peut en même temps établir sa propre vitesse de communication souhaitée avec l'unité A. L'unité B peut initialement informer l'unité A qu'elle est prête à recevoir des UDP jusque et y compris le numéro d'ordre 10. Lorsque ces dix UDP sont reçues, l'unité B peut en demander une autre dizaine, en portant le numéro d'ordre à 20, ou peut ralentir ou accélérer la vitesse de communication en

choisissant un numéro d'ordre inférieur ou supérieur. Les numéros d'ordre initiaux sont établis pendant une phase de communication impliquant une connexion. Les numéros d'ordre sont ensuite tenus à jour en tant que parties du signal d'accusé de réception de message (ACK) qui est renvoyé du destinataire à l'initiateur.

A titre de protection contre un arrêt du système qui se produirait si les deux unités de communication attendaient des accusés de réception mutuels, les entités de la couche de transport dans chaque unité sont conçues ou sollicitées pour émettre de façon routinière des signaux d'accusé de réception redondants qui acheminent les messages de données autorisés en cours qui peuvent être reçus. Ces informations autorisées sont qualifiées de fenêtre.

Pour mieux comprendre le système d'accusé de réception positif mis en oeuvre par MAP, on peut se référer à la Fig. 4 qui illustre un exemple de communication entre un initiateur et un destinataire. Il convient de ne pas oublier que MAP met en oeuvre un système de communication duplex intégral, les deux unités de communication pouvant être simultanément initiatrices et destinataires. Deux colonnes sont illustrées sur la Fig. 4 et sont désignées par les vocables initiateur et destinataire. Dans ces colonnes figure une séquence de transmissions de messages portant des flèches indiquant le sens de la transmission. La communication débute par une séquence d'établissement de connexion. La partie d'établissement de connexion de la communication est indiquée d'une manière générale en 40. La séquence débute en 42 par l'envoi par l'initiateur d'une requête de connexion (RC) au destinataire. Le destinataire répond en transmettant un message de confirmation de connexion (CC). L'initiateur accuse alors réception de la confirmation

de connexion en émettant un accusé de réception (ACK). Pendant cette phase d'établissement de la connexion, l'initiateur et le destinataire spécifient les numéros d'ordre minimum et maximum qu'ils sont prêts à recevoir (à savoir la fenêtre). A titre d'exemple, seules des communications initiées par la première unité et reçues par la seconde sont illustrées. Il va de soi qu'un diagramme semblable serait obtenu si les communications initiées par la seconde unité en vue d'être reçues par la première étaient représentées.

Lorsque la connexion est initialement établie, on suppose que la communication débute par le numéro d'ordre 0 (le premier message d'information d'utilisateur de transport). Le numéro d'ordre admissible le plus élevé établi par le destinataire, ainsi que le numéro d'ordre initial 0, déterminent le nombre de messages pouvant être envoyés au destinataire. Ce nombre peut avantageusement être représenté par une fenêtre illustrée sur la Fig. 5 définie entre la dernière requête (dans ce cas le numéro d'ordre initial 0) et le numéro d'ordre le plus élevé admissible requis. Si, par exemple, le destinataire souhaite limiter la réception à 100 messages, le bord inférieur 44 de la fenêtre se situera initialement au numéro d'ordre 0 tandis que son bord supérieur 46 sera au numéro d'ordre 100.

Comme le montre la Fig. 4, dès que l'établissement de la connexion en 40 est effectué, l'initiateur entame la transmission de données UDP (TD). A titre d'exemple, on suppose que le destinataire a requis l'émission de trois messages (UDP). Cela étant, l'initiateur transmet les trois premiers messages correspondant aux numéros d'ordre 0, 1 et 2. Ceci est indiqué d'une manière générale en 48. Lorsque le destinataire reçoit le troisième message, il renvoie un signal

d'accusé de réception (ACK) à l'initiateur. Ceci est indiqué d'une manière générale en 50. Le signal d'accusé de réception contient le numéro d'ordre le plus élevé reçu, dans ce cas celui du message 2. L'initiateur sait donc que le destinataire a reçu les trois paquets. S'il le souhaite, le destinataire pourrait, avec son accusé de réception, modifier sa dimension de fenêtre en demandant un groupe supplémentaire de paquets différant du nombre initialement requis. Par exemple, le destinataire peut souhaiter permettre la transmission d'un nombre de messages allant jusqu'à 10. Dans ce cas, le destinataire devrait revoir son plus haut numéro d'ordre admissible pour le porter à 13. Si le destinataire ne procède pas à la modification de son plus haut numéro d'ordre admissible, ce numéro reste à la valeur précédemment choisie, c'est-à-dire 3, et aucun autre message ne peut être transmis.

A supposer que des messages supplémentaires ont été demandés, l'initiateur continue à émettre des messages et s'attend à recevoir ensuite des accusés de réception. Ceci est indiqué en 52 pour les messages 3 et 4. Il est évidemment possible qu'un message particulier ne soit pas reçu dans l'ordre adéquat. Un paquet donné peut prendre temporairement du retard ou être perdu pendant une transmission. En 54, l'initiateur émet des messages 5 et 6. Le destinataire n'accuse réception que du message numéro 5 ce qui illustre le cas dans lequel le message numéro 6 n'est pas reçu dans l'ordre. Pendant ce temps, l'initiateur continue à émettre des messages 7 et 8, par exemple en 56. Le destinataire, cependant, ne peut pas accuser réception des messages numéros 6, 7 et 8 puisqu'il n'a pas encore accusé réception du message numéro 5.

Suivant l'opération standard du protocole de

couche de transport, le destinataire doit émettre périodiquement un accusé de réception reflétant le dernier numéro d'ordre qu'il a reçu. Dans ce cas, le destinataire a reçu le message numéro 5. Cela étant, lors du pas 58, le destinataire émet un accusé de réception périodique pour la réception du numéro de message 5. L'initiateur détermine, à partir de cet accusé de réception, que le destinataire n'a pas encore reçu le message numéro 6 (émis précédemment) en 54. Cela étant, l'initiateur réémet le message numéro 6, en 60, et le destinataire répond alors par un accusé de réception du message numéro 8. Dans cet exemple, on présume que seul le message numéro 6 n'a pas été reçu par le destinataire et que les messages numéros 7 et 8 ont été reçus et ont été stockés dans l'attente de la réception du message numéro 5.

Il est important, pour comprendre l'invention, de reconnaître que la fenêtre définie par les bords supérieur et inférieur illustrés sur la Fig. 5 peut changer de dimensions à chaque accusé de réception. Le bord inférieur 44 de la fenêtre représente le dernier numéro d'ordre ayant fait l'objet d'un accusé de réception et le bord supérieur de la fenêtre représente le numéro d'ordre maximum permis. L'invention a trait principalement à la régulation du flux de données entre les entités de la couche de réseau du niveau trois au moyen de mécanismes de la couche de transport du niveau quatre. Pour être plus complet, on décrira brièvement les couches des niveaux cinq, six et sept.

La couche de niveau cinq ou de session illustrée sur la Fig. 1 est destinée principalement à synchroniser les unités de communication pour qu'elles se relayent dans l'établissement d'une communication. La couche de session est grossièrement analogue à une ponctuation de phrase, qui divise une idée complète,

permettant à un autre message, éventuellement de l'autre unité, de répondre.

Le sixième niveau ou couche de présentation est responsable du traitement de la représentation des données entre des unités de communication. Des unités de communication ne peuvent communiquer qu'au moyen de langages codés semblables ou de fichiers de types semblables. La couche de présentation fournit un accord sur un langage, codage ou type de fichier commun par lequel les deux systèmes incompatibles peuvent communiquer. La couche de présentation peut aussi effectuer le codage et le décodage des données en vue du chiffrement des données et d'opérations analogues. Quoique le modèle OSI permet à des systèmes de communication de négocier un protocole à ce niveau, la spécification MAP appliquée pour l'instant n'utilise pas une telle négociation.

Le niveau final et septième est la couche d'application. La couche d'application définit, pour certains contextes, le langage que l'on va utiliser pendant la communication. Le contexte peut être implicite, sur base de la nature de la communication, ou être explicite. La couche d'application est nécessaire parce que les données communiquées peuvent être organisées selon des manières fondamentalement différentes par des systèmes informatiques différents. La couche d'application définit un mode de référence aux choses, qui a fait l'objet d'un accord mutuel. L'une ou les deux unités de communication peuvent devoir transposer leurs données dans le format qui a fait l'objet de l'accord mutuel, si un tel format n'est pas essentiellement prévu.

Comme expliqué plus haut, la présente invention s'attaque principalement au test de la conformité de l'entité de protocole interréseau de la couche de

réseau de niveau trois d'un système sous test. Le but est de transmettre des données valides et invalides à l'entité de la couche de réseau afin de déterminer si elle fait une distinction adéquate entre les deux. Un autre but est d'amener la couche de réseau à faire monter des données vers la couche de transport et à recevoir des données descendant de la couche de transport, sans erreur. La couche de réseau est sans connexion en ce sens qu'elle n'applique pas de protocoles à connexion comme certaines des couches supérieures le font. Avant l'invention, pour tester la couche de réseau, il fallait placer un programme de test spécial dans la couche de transport du système sous test (ou au niveau de l'interface de la couche de réseau et de la couche de transport) en vue d'émettre des messages de test prédéfinis à travers l'interface transport/réseau vers la couche de réseau afin de voir si cette couche de réseau les reçoit convenablement. Le programme de test doit aussi réagir à des messages de test prédéfinis émis à travers l'interface transport/réseau depuis la couche de réseau afin de déterminer si cette couche de réseau achemine convenablement des messages. L'addition de ce logiciel de test augmente la complexité du système de communication et rend difficile la conception d'un logiciel de haut rendement qui peut exiger l'intégration de certaines fonctions étroitement apparentées des couches de transport et de réseau.

Après avoir ainsi présenté une description générale de l'invention et de son milieu opérationnel actuellement préféré, on donnera ci-après une explication plus détaillée du système de test intégré.

L'invention se réfère au système ou à la technique de "test intégré" parce que l'interface de service de réseau dans le système sous test peut rester couplée naturellement à la couche de transport. Le

procédé de test commun dans lequel l'interface de service de réseau doit être découplée de la couche de transport et doit être pilotée par un logiciel spécialisé dans le système sous test est qualifié de système ou technique de test "exposé".

L'invention utilise comme base d'application, le logiciel qui a été mis au point pour exécuter la technique de test exposé. De plus, l'invention est à même d'exécuter la même série de jeux d'essai que le système de test exposé. Le choix de la technique est laissé au concepteur du système sous test. En d'autres termes, le système de test "intégré" utilise le logiciel de système de test interréseau "exposé" existant comme base d'application. En fait, un seul système de test est maintenu, un ordre de l'utilisateur étant nécessaire pour la sélection si un contrôle de service intégré ou exposé est disponible.

La version exposée continue à utiliser les unités de données de protocole de gestion de test (UDPGT) définies par le NBS (Bureau National de Normalisation) et le testeur supérieur associé pour le contrôle de l'interface de service IST. Une description du protocole GT (gestion de test) peut être trouvée dans ISCT/SNA-85-7, National Bureau of Standards, "The Design of a Test System for Implementations of ISO Connectionless Network Protocol", juillet 1985 (NBS 86).

La Fig. 8 illustre le procédé actuellement préféré de mise en oeuvre du système de test intégré conforme à l'invention. La séquence débute au pas 100 et procède à l'établissement de la communication à connexion entre le logiciel du système de test et celui du système sous test en 102. Le reste de la séquence de test illustrée dans l'organigramme de la Fig. 8 comprend plusieurs points de branchement (en 104, 110,

120, 122, 128, 136 et 144) où des tests spécifiques peuvent être exécutés.

Par exemple, si on souhaite tester le transmetteur de données de protocole interréseau (PI), un branchement est prévu en 104 pour exécuter les pas 106 et 108. Dans le pas 106, une unité de données prédéfinie, qualifiée de UDSR, contenant une UDPT-AK, est envoyée avec un crédit d'une unité en plus que le numéro d'ordre précédent. Ceci ouvre la fenêtre de crédit pour permettre le renvoi d'une unité de message à partir du système sous test. Lors de l'implémentation de ces tests, le système sous test doit être normalement préconditionné pour envoyer un grand nombre d'octets de données et attendre leur réception. En ouvrant la fenêtre de crédit pour un crédit d'une unité en plus que le numéro d'ordre précédent, on permet à l'un des divers octets de données de passer du système sous test au système de test. Dans le pas 108, le système de test indique par un drapeau qu'il s'attend à recevoir une UPDT-TD présentant un numéro d'ordre supérieur d'une unité au numéro précédent.

Si l'on souhaite tester le récepteur de données PI, le pas 110 passe la commande au pas 112 qui envoie une UDSR contenant une UPDT-TD portant un numéro d'ordre supérieur d'une unité au numéro d'ordre précédent. Une décision a été prise dans le pas 114 au sujet de l'altération éventuelle des UDPI qui véhiculent la UDSR. En cas d'altération, un drapeau est affiché pour avertir le système de test de s'attendre à un message d'erreur UDPI-ER ou à une absence de réponse. Si les données ne sont pas altérées, un drapeau est affiché pour avertir le système de test de s'attendre à un message d'accusé de réception UDPT-AK portant un numéro d'ordre supérieur d'une unité au numéro d'ordre précédent.

Comme expliqué plus haut, le protocole de communication map exige l'envoi périodique de signaux d'accusé de réception doubles à titre de protection contre un arrêt du système. Ces signaux d'accusé de réception doubles sont interceptés au point de branchement 120. Si un signal d'accusé de réception double est reçu, aucune action n'est entreprise par le système de test et la commande du déroulement revient simplement à la partie supérieure de la boucle.

Pour vérifier si la séquence d'accusé de réception affichée dans le pas 18 est reçue, le pas 122 contrôle la réception de l'accusé de réception attendu et passe au pas 124 si l'accusé de réception est reçu. Le pas 124 vérifie l'accusé de réception pour voir s'il a été émis en réponse au test d'émetteur et si, dans l'affirmative, aucune panne n'est signalée, sinon une panne est signalée en 126.

Un test semblable est effectué au pas 128 pour déterminer si l'unité UDPT-TD signalée au pas 108 a été reçue. Dans l'affirmative, le pas 130 intervient pour déterminer si cette réception s'est effectuée en fonction d'un test de récepteur. Dans l'affirmative, un signal d'accusé de réception UPDT-AK est envoyé en 132. Sinon, un état de panne est signalé en 134.

D'une manière analogue, au pas 136, un test est effectué pour déterminer si le message d'erreur attendu affiché dans le pas 116 est reçu. Dans l'affirmative, il faut déterminer si ce signal d'erreur UDPI-ER est obtenu en réponse à un test de récepteur. Dans l'affirmative, une UDPT-TD double est émise, sinon un état de panne est signalé en 142.

Finalement, si d'autres UDPI non-TD ou non-ER UDPT sont reçues ou si des UDPI-TD incomplètement réassemblées sont reçues, un branchement est prévu en 144 pour signaler une panne en 146. En 148 est prise la

décision de terminer le test ou de revenir à la partie supérieure de la boucle en vue de poursuivre le test. Si le test se termine, la connexion est interrompue en 150, après quoi le programme se termine en 152.

La version intégrée utilise des unités de données de protocole de transport standard (UDPT) et une entité de transport conforme au protocole MAP 2:1 pour la commande de l'interface de service de réseau IST. L'entité de transport IST est, à son tour, commandée par un interpréteur de scénario à distance utilisé dans le test du transport. Un seul scénario de transport (jeu d'essai) est requis pour un test IP intégré. Aucun accès direct à l'interface de service de réseau de l'IST n'est ainsi requis.

L'utilisation contrôlée du protocole de transport classe 4 MAP/ISO dans des tests intégrés offre la même fonctionnalité pour contrôler et observer l'activité de l'interface de service de réseau en tant que protocole sur commande dans le procédé exposé. (Il y a des exceptions mineures, mais il est inutile de débiter par la fonctionnalité GT perdue). Pendant l'exécution du jeu d'essai de protocole interréseau (PI), seules des unités UDPT-TD et UDPT-AK sont transportées dans les portions de données UDPI. L'établissement et la terminaison des connexions de transport sont effectués en dehors de l'exécution d'un quelconque cas de test PI.

L'établissement d'une connexion est amorcé par émission d'un ordre "connecter". Une UDPT RC-CC-AK échangée est attendue. Le système de test continue à transmettre des RC pendant un laps de temps raisonnable avant d'abandonner. Des jeux d'essai PI ne sont exécutés que si une connexion s'avère ouverte.

Des valeurs de synchroniseur de transport et paramétriques sont établies par négociation RC-CC ou

sont prédéfinies dans le plan de test PI intégré. Ces valeurs seront vraisemblablement semblables aux valeurs utilisées dans des tests de conformité de protocole de transport en vigueur.

Des données de UDPI valides, provenant du testeur inférieur, sont des UDPT-TD dans l'ordre correct avec un format de données respectant les prévisions de l'interpréteur de scénario à distance. L'interpréteur de scénario à distance (IDS) est le même que celui utilisé dans le test de protocole de transport standard. Les données UDPI renvoyées dans des UDPI invalides sont également envoyées sous la forme de UDPT-TD disposées dans un ordre correct. Si la UDPI invalide est acceptée, le système de test signale une panne. Si une réponse d'erreur correcte est renvoyée, le test se poursuit sans problème. Pour maintenir un ordre correct des messages de transport, la UDPT-TD est retransmise en une UDPI valide après réception de la réponse d'erreur correcte.

Une UDPT-AK à bord de fenêtre inférieur correctement accru est attendue en réponse à une UDPT-TD véhiculée dans une UDPI valide.

Pour empêcher l'entité de transport IST d'envoyer prématurément des UDPT-TD, les UDPT-AK émises par le système de test ne fournissent pas de crédit. Lorsqu'une UDSR (unité de données de service de réseau, c'est-à-dire une UDPT-TD dans un ordre correct) de l'IST est souhaitée, une UDPT-AK accordant un crédit est émise par le testeur inférieur. Si le testeur inférieur reçoit une UDSR valide (c'est-à-dire une UDPT-TD valide), la fenêtre est refermée et la UDPT-TD fait l'objet d'un accusé de réception. Cette UDPT-AK accordant du crédit ne spécifie pas la dimension de la UDSR. Une telle spécification paraît être indésirable, car elle s'efforcerait de régir un paramètre d'implé-

mentation spécifique. Par conséquent, pour un test de conformité, l'accusé de réception AK avec crédit est suffisant. Il est à noter que les données UDPT-TD provenant de l'IST paraissent suivre le format correct de l'interpréteur de scénario à distance.

Le testeur inférieur et l'IST doivent transmettre périodiquement une UDPT-AK pour satisfaire l'exigence de temps de la fenêtre de transport. Ceci permet aux deux parties d'avoir connaissance du maintien de leur viabilité réciproque.

Contrôle de service et observation intégrés.

Le système à mode exposé utilise une entrée de conformité pour garder la trace de toutes les UPDI attendues de l'IST. Cette table est convenablement remplie lorsque le système de test émet une UDPI et l'entrée de la table est éliminée après réception d'une UPDI satisfaisante.

Dans le mode intégré, cette table n'est pas utilisée dans des processus de connexion et de déconnexion. Pour des processus de connexion et de déconnexion, il suffit de consulter la structure de données du contexte de la connexion de transport.

Pendant l'exécution du jeu d'essai PI, les entrées de la table de conformité et le contexte de transport sont utilisés. Etant donné que seules des UDPT-TD et UDPT-AK sont utilisées pendant l'exécution du jeu d'essai PI, cette exécution s'effectue d'une manière directe. Le système de test utilise les règles suivantes.

1. Réception d'une UDPT-TD : si la TD s'effectue dans l'ordre, elle se situe dans la fenêtre ouverte, la partie de données est correctement formatée et une entrée de conformité engendrée par une UDPT-AK avec crédit a été envoyée, l'entrée est résolue, le "Tctx" est mis à jour et une UDPT-AK est émise par le

testeur inférieur.

2. Réception d'une UDPT-AK à numéro d'ordre double : mise à jour du "Tctx" d'une manière appropriée.

3. Réception d'une UDPT-AK à numéro d'ordre supérieur : si une entrée de conformité de type "UDPT de données valides" existe et qu'une UDPT-TD marquante a été transmise, l'entrée est résolue et le "Tctx" est mis à jour.

4. Envoi d'une UDPT-TD dans une UDPI de données valides : une bonne UDPI-TD de n octets de données selon le jeu d'essai doit être envoyée. Une entrée de conformité de type "UDPI de données valides" est produite et l'UDPI est envoyée.

5. Envoi d'une UDPT-TD dans une UDPI de données invalides : une mauvaise UDPT-TD de n octets de données doit être envoyée et un drapeau indicateur d'erreur est affiché, selon le jeu d'essai. Une entrée de conformité de type "UDPI de données valides" est produite et l'UDPI est émise.

6. Envoi de UDPT-AK avec numéro d'ordre double : le "Tctx" indique que le synchroniseur de fenêtre a terminé.

7. Envoi de UDPT-AK avec numéro d'ordre double et crédit : une UDSR portant une UDPT-TD à numéro d'ordre correct est attendue. Une entrée de conformité est établie et l'UDPI de données avec la UDPT-AK est envoyée.

8. Envoi de UDPT-AK avec numéro d'ordre supérieur : une entrée "UDPI de données valides" a été résolue et une UDPI de données est envoyée.

9. Réception de UDPT-ER : si une entrée de conformité "UDPI-ER" existe, l'entrée est résolue.

Intégration avec logiciel interréseau en vigueur.

La fonctionnalité du test intégré est combinée avec le logiciel existant d'une manière telle que les modifications soient isolées et bien structurées. Un examen de la structure logicielle de système exposé est présenté et est suivi des modifications apportées pour soutenir le système intégré.

Le logiciel de système exposé est structuré comme un processus de bouclage continu. Les modifications/altérations destinées à soutenir le système intégré opèrent dans cette structure. Le pseudocode suivant décrit le système à deux modes. Les instructions du type ajoute sont marquées "*A" celles du type modification sont marquées "*M".

Pour toujours

[

```

    si (ordre du terminal d'utilisateur)
*M      exécuter ordre;
*A      si (mode intégré)
*A
*A      envoi de UDPT de maintien de connexion;
*A      envoi de AK pour confirmer les TD valides;
    si (le jeu d'essai a été mis en file d'attente et
        qu'aucun jeu d'essai n'est en cours, lire le
        jeu d'essai;
*M      construction d'UDPI abrégées
        mise en file d'attente des UDPI abrégées en
        vue de la transformation;
*M      création d'entrées de conformité pour la
        réception;
        jeu d'essai considéré comme étant en cours;
    si (les UDPI abrégées sont mises en file
        d'attente);
        transformation des UDPI abrégées en UDPI
        concrètes;
```

```

mise en file d'attente des UDPI concrètes en
vue de l'envoi;
si (les UDPI concrètes sont mises en file
d'attente)
envoi des UDPI concrètes;
si (les UDPI arrivantes se trouvent dans la file
d'attente de réception)
valider l'en-tête d'UDPI;
*M vérifier données et les réassembler;
*M évaluer la conformité au jeu d'essai;
si (événement de synchroniseur du jeu d'essai)
intervention appropriée;
si (test en cours et toutes les entrées de
conformité résolues)
le test est terminé;

```

]

Les modifications destinées à soutenir le système intégré sont les suivantes.

1. La structure de données globale "Tctx" fournit les données nécessaires pour entretenir le fonctionnement de la connexion de transport. Ces données comprennent les variables nécessaires pour engendrer et analyser des données pour l'interpréteur de scénario à distance.

2. Les données UDPI engendrées pendant le temps de formation des UDPT abrégées sont fournies par une fonction appelée gen-data(). Cette fonction est modifiée pour produire des UDPT-TD en mode intégré. Des modifications mineures à la logique créant les entrées de conformité sont requises.

3. Des UDPI sont produites pour demander à l'IST d'émettre des données. La partie de données de ces UDPI véhicule des GTUDP "émission de UDSR" dans le mode exposé. Ces GTUDP sont engendrées pendant le temps de formation des UDPI abrégées par la fonction "udsr-

données()". Cette fonction est modifiée pour le mode intégré en vue de produire des UDPT-AK à crédit(s). Des modifications mineures à la logique de création des entrées de conformité sont requises.

4. Les parties de données des UDPI reçues sont analysées par plusieurs sous-programmes différents en mode exposé. Ces sous-programmes dépendent du fait que les données identifiées constituent une GTUDP ou simplement des données mises en ordre. Ces sous-programmes sont appelés à partir de la fonction "récepteur()" ou "réassembleur()".

5. Dans le mode intégré, un seul sous-programme intitulé "analyse de données()" est appelé. Ce sous-programme est appelé depuis l'intérieur du "réassembleur()". Toutes les parties de données UDPI sont passées au "réassembleur", que le réassemblage soit requis ou non (il s'agit là de la pratique existante pour le mode exposé également). Le sous-programme "analyse de données()" est modifié pour appliquer une analyse du type UDP de transport. Les TD reçues sont attendues dans la fenêtre admissible. Les AK reçus sont attendus comme des accusés de réception ne portant que sur des TD marquantes ou comme des doubles pour le maintien de la connexion. La fonction "analyse de données()" renvoie un compte des erreurs détectées et le "réassembleur" imprime un message indiquant des violations de codage ou de longueur.

6. La connexion de transport est gérée par le sous-programme "transport()". Ce sous-programme est appelé dans la boucle "pour toujours" après l'appel du traitement des ordres d'utilisateur. Les ordres de connexion, de déconnexion et de fermeture de l'utilisateur (s'ils sont valides) établissent des variables "Tctx" indiquant qu'un service souhaité doit être réalisé". Le sous-programme de "transport" envoie des

UDPI concrètes requises quelconques directement. Des UDPI reçues sont traitées par le sous-programme "analyse de données()" dans le "réassembleur()". Le sous-programme "analyse de données()" base sa vérification sur les indicateurs dans "Tctx".

Analyse de conformité.

Le mécanisme d'analyse de conformité existant est utilisé presque sans modification pour le mode intégré. Les modifications prévues comprennent (1) le fait que la variable de numéro d'ordre dans chaque entrée de conformité est réglée de façon à être en corrélation avec l'activité UDP de transport; et (2) le fait qu'une entrée peut exiger plus d'une réponse à résoudre. Ceci implique que toutes les erreurs existantes dans le mécanisme d'analyse de conformité n'ont pas encore été corrigées.

Les erreurs de niveau de transport sont annoncées par des messages d'erreur parvenant à la console de l'opérateur. C'est l'opérateur qui doit décider si un message d'erreur indique une activité sans conséquence, une non-testabilité, ou une non-conformité.

Usage de la connexion de transport.

La connexion de transport est toujours amorcée par le système de test. Le LT propose des demandes de bas niveau à la plupart des négociations, ce qui ne donne pas le choix à l'IST. Ces demandes sont : totaux de contrôle actifs, format normal, pas de données accélérées. La dimension maxima proposée pour une UDPT est de 512 octets.

La valeur de dimension est choisie de manière à prendre en charge les limitations de matériel en vigueur du système de test. L'IST peut négocier cette valeur vers le bas. Quelle que soit la dimension maxima négociée, le système de test n'envoie jamais de UDPI

supérieure à la dimension maxima du message dans le système de test (pour l'instant 600 octets). La série en vigueur de jeu d'essai PI n'exige pas qu'une UDSR supérieure à 512 octets soit envoyée ou reçue par le système de test. Le système de test amorce également la déconnexion lorsque la console le lui ordonne.

Chaque accusé de réception AK transmis transporte le paramètre de confirmation de commande de flux (ccf). La réception d'accusés de réception AK nus (sans paramètre) provoque une retransmission du dernier AK.

Structure de données.

La structure de données suivante est utilisée pour maintenir le contexte du transport.

```

struct TOONTEXT Tctx
{
    int state; /* major state */
    int TCref; /* LT reference number */
    int RTref; /* IUT reference number */
    int sizebid; /* maximum negotiated tpdu size */
    int send-seq; /* sending sequence number */
    int send-ssq; /* sending subsequence number */
    int send-cdt; /* sending credit */
    int rcv-seq; /* receiving sequence number */
    int rcv-ssq; /* receiving subsequence number */
    int rcv-cdt; /* receiving credit */
    int window time; /* window timer value */
    int inact-time; /* inactivity timer value */
    int giveup-time; /* giveuptimer value */
    int retran-time; /* retransmission timer value */
    int retran-count; /* retransmission count */
    int cr-len; /* cr pdu length */
    int ak-len; /* ak pdu length */
    int dt-hlen; /* dt pdu header length */
    int dt-tlen; /* dt pdu total length */
    int dr-len; /* dr pdu length */
    int dc-len; /* dc pdu length */
    int ip-hlen; /* ipdu header length */
    int ip-dui; /* next dui to use */
    unsigned char crpdu[512]; /* cr tpdu for sending */
    unsigned char akpdu[512]; /* ak tpdu for sending */
    unsigned char dtpdu[512]; /* dt tpdu for sending */
}

```

```

unsigned
  char drpdu [512]; /* dr tpdu for sending */
unsigned
  char dcpdu [512]; /* dc tpdu for sending */
unsigned
  char ippdu [700]; /* data ipdu for sending */
long send-count; /* count of transport data sent */
long send-goal; /* total amount of data to be sent */
long recv-count; /* count of transport data received */
long recv-goal; /* total amount of data to be received */
int akstat; /* classification of ak type */
int send-dtseq; /* sequence number of last dt sent */
int send-akseq; /* sequence number of last ak sent */
int send-eot; /* eot sent */
int recv-eot; /* eot received */
int timebase; /* time at program initialization */
int send-time; /* time last data transmitted */
int recv-time; /* time last data received */
int send-vector; /* vector indicating next pdu to send */
int reason; /* disconnect reason */
int dest; /* out-of-context pdu dest */
int src; /* out-of-context pdu src */
int TCsent roll; /* count of dt sent sequence number rollover */
int TCack roll; /* count of ak sent sequence number rollover */
int RTSent roll; /* count of dt recv sequence number rollover */
int RTack roll; /* count of ak recv sequence number rollover */
char *fault; /* pointer to pdu error */
struct
  DTREG DTreg [10]; /* registers to buffer sent, but unacked dts */
struct
  INPDU *pduptr; /* pointer to received pdu */

```

Maintien de la connexion.

Des UDPT de transport sont formées pour l'envoi à deux fins différentes. Elles sont requises, soit pour former la partie de données UDPI définie par un jeu d'essai PI, soit pour établir, maintenir ou terminer une connexion de transport.

Les UDPT utilisées pour le support de jeu d'essai PI sont produites au moment de la formation de l'UDPI abstraite. La construction de ces UDPT (uniquement des TD ou AK) se base sur la structure "Tctx" pour l'obtention de valeurs appropriées.

Les UDPT envoyées pour le maintien de la connexion sont envoyées directement (les files

d'attente d'envoi normales sont dérivées). Ces UDPT sont envoyées après traitement des ordres de console. L'en-tête de l'UDPI est prédéfinie au moyen du IUD (identificateur d'unité de données), de l'indicateur de longueur de segment, de l'indicateur de longueur totale et du total de contrôle d'en-tête convenablement modifié. Toutes ces informations sont présentes dans le "Tctx". Les valeurs IUD démarrent à (décimale) 1000, de manière à ne pas être en conflit avec les IUD de jeu d'essai PI. Les jeux d'essai PI doivent éviter de spécifier des UDPI présentant des valeurs IUD supérieures à 1000 pour éviter tout risque de conflit. Etant donné que tous les jeux d'essai PI en vigueur utilisent des IUD débutant à (décimale) 1 (ou pour des tests de jeux partiels pas de IUD du tout), les UDPI utilisées pour la commande peuvent être facilement distinguées de UDPI sous test).

Toutes les UDPI reçues (les deux modes) sont attaquées par l'intermédiaire de la fonction "analyse de données". Dans le mode exposé, la UDPI et ses données sont évaluées pendant le réassemblage et une structure de données appelée "SEGMT" du type "REASM-SEG" est créée pour maintenir de l'information pertinente au sujet du segment. Le tampon retenant le segment effectif est alors libéré. Le procédé de réassemblage maintient une liste enchaînée de structures "segmt" à utiliser dans le réassemblage.

Le type de données "REASM-SEG" est modifié pour soutenir le mode intégré par le maintien d'informations supplémentaires se rapportant aux UDPT. Ces informations comprennent des UDPI partiellement réassemblées. Si les données d'une UDPI sont valides selon les règles du protocole de transport et les buts de l'interpréteur de scénario, seule l'entrée de conformité doit être respectée. Cette entrée est

respectée si le numéro d'ordre de la transmission de données TD ou de l'accusé de réception AK arrivant est en corrélation avec celui de l'entrée de conformité.

Les accusés de réception arrivant en double ramènent simplement à zéro le temps d'inactivité et sont rejetés. Si le double est "nu", un drapeau est affiché dans "Tctx" indiquant qu'un accusé de réception AK doit être transmis. Tous les accusés de réception transmis contiennent le paramètre ccf. Tous les autres TD et AK déclassés sont identifiés comme étant des erreurs.

La transmission de UDPT est déclenchée par des drapeaux qui sont affichés dans "Tctx". Dans la boucle principale, immédiatement après la recherche d'ordres éventuels de la console, la fonction "transport()" est appelée. Si des drapeaux sont affichés, une UDPI appropriée est envoyée. Le mécanisme est utilisé pour envoyer toutes les UDPT, à l'exception des TD et des AK qui sont requis comme données UDPI pendant un jeu d'essai PI.

Des UDPT modèles sont conservées dans la structure "Tctx". Une telle UDPT est copiée dans un tampon contenant une en-tête UDPI et les paramètres appropriés sont modifiés dans l'en-tête UDPI et dans l'UDPT. Ce tampon est ensuite passé au sous-programme d'écriture dérivant la structure de file d'attente utilisée pour les UDPI produites par les jeux d'essai PI.

Le sous-programme "transport" envoie les UDPT nécessaires pour répondre à un ordre de l'utilisateur (connecter, fermer, déconnecter), accuser réception d'une TD valide et fermer une fenêtre (qui était précédemment permise par un accusé de réception AK produit par un jeu d'essai PI), et pour satisfaire des synchroniseurs de transport standard.

R E V E N D I C A T I O N S

1.- Procédé pour tester la conformité d'une implémentation d'un protocole interréseau dans un système sous test à l'égard de sa spécification, caractérisé en ce que :

on utilise les mécanismes de réglage de flux et d'accusé de réception du protocole de transport pour régir l'entité de protocole interréseau (PI) dans le système sous test.

2.- Procédé suivant la revendication 1, caractérisé en ce que les mécanismes de réglage de flux et d'accusé de réception sont utilisés pour maintenir un environnement de test réglé.

3.- Procédé suivant la revendication 1, caractérisé en ce que les mécanismes de réglage de flux spécifient la quantité de messages vraisemblablement transmise par l'entité de protocole interréseau dans le système sous test.

4.- Procédé suivant la revendication 1, caractérisé en ce que le mécanisme d'accusé de réception est utilisé pour déterminer si l'entité de protocole interréseau dans le système sous test reçoit correctement des données et les transmet à un niveau d'utilisateur supérieur dans le système sous test.

5.- Procédé suivant la revendication 1, caractérisé en ce que le mécanisme de réglage de flux est utilisé pour définir une fenêtre pour des transmissions admissibles dans l'entité de protocole interréseau dans le système sous test.

6.- Procédé pour tester la conformité d'une implémentation d'un protocole interréseau dans un système sous test à l'égard de sa spécification, caractérisé en ce que :

on utilise le mécanisme de réglage de flux du

protocole de transport pour régir l'entité de protocole interréseau (PI) dans le système sous test.

7.- Procédé pour tester la conformité d'une implémentation d'un protocole interréseau dans un système sous test à l'égard de sa spécification, caractérisé en ce que :

on utilise le mécanisme d'accusé de réception du protocole de transport pour régir l'entité de protocole interréseau (PI) dans le système sous test.

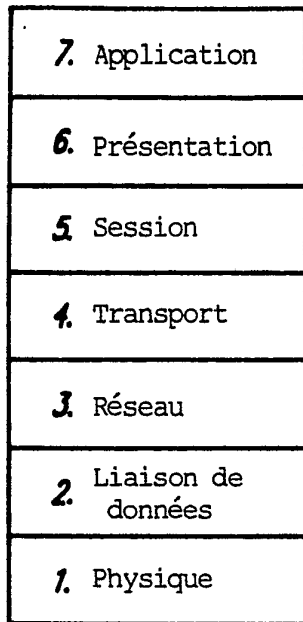


FIG. 1.

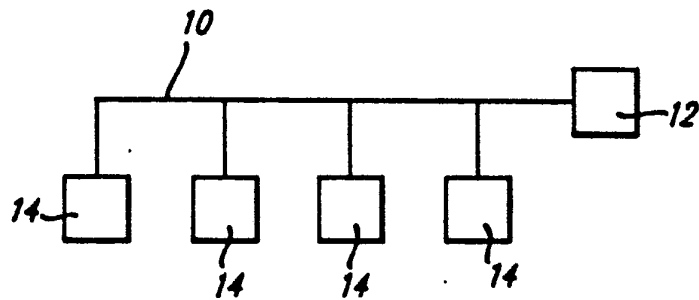


FIG. 2.

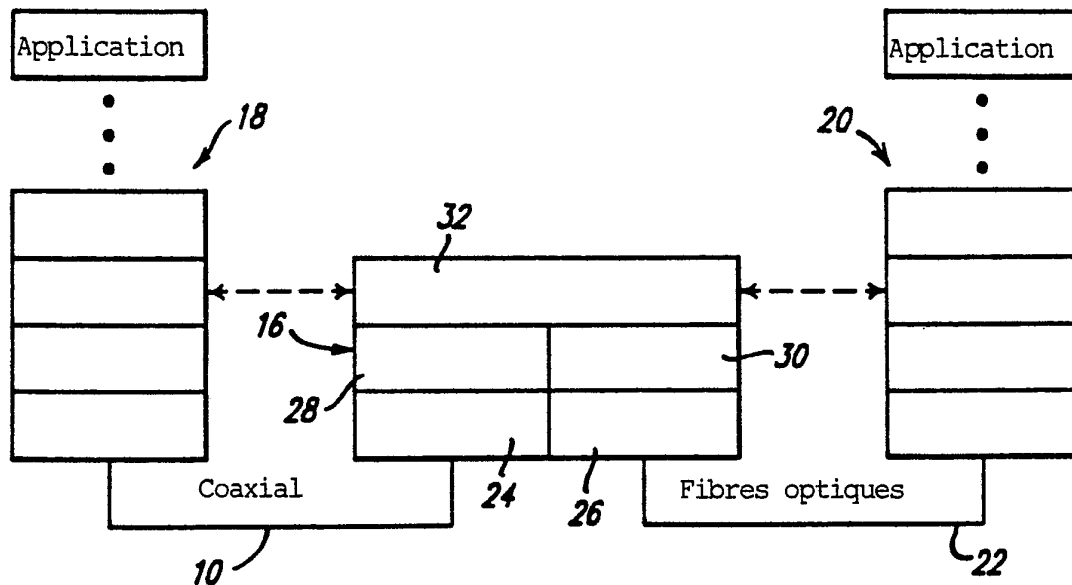
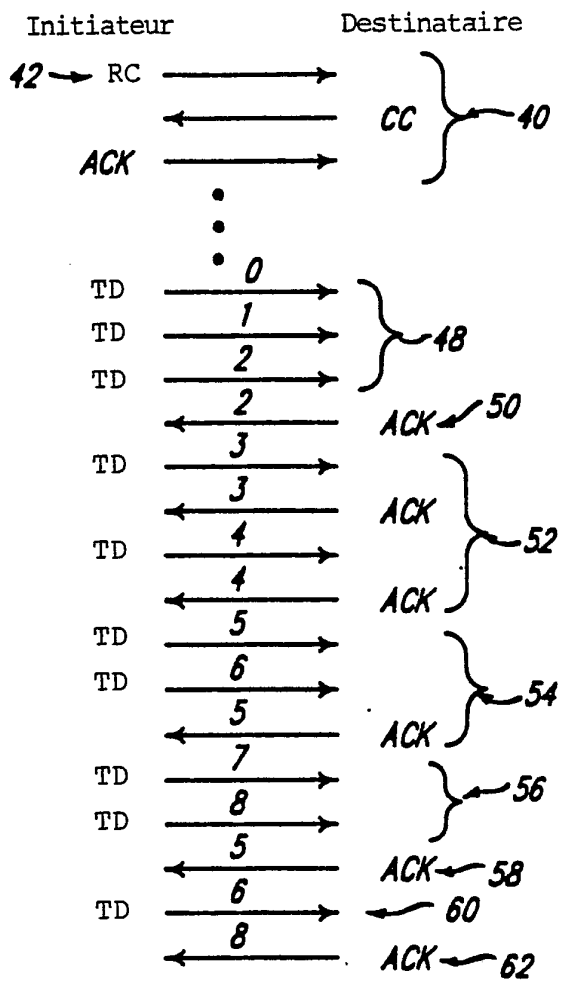


FIG. 3.



III. 4.

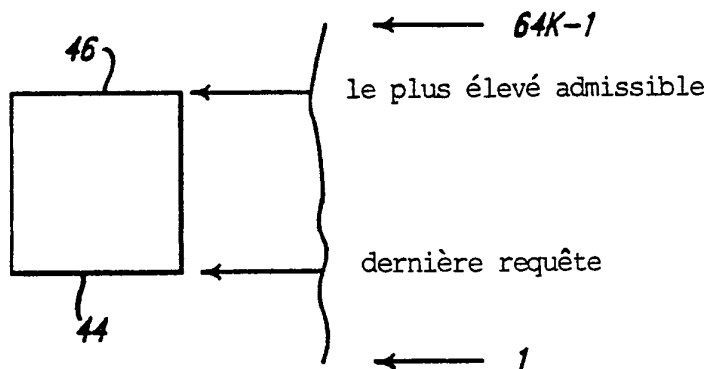


FIG. 5.

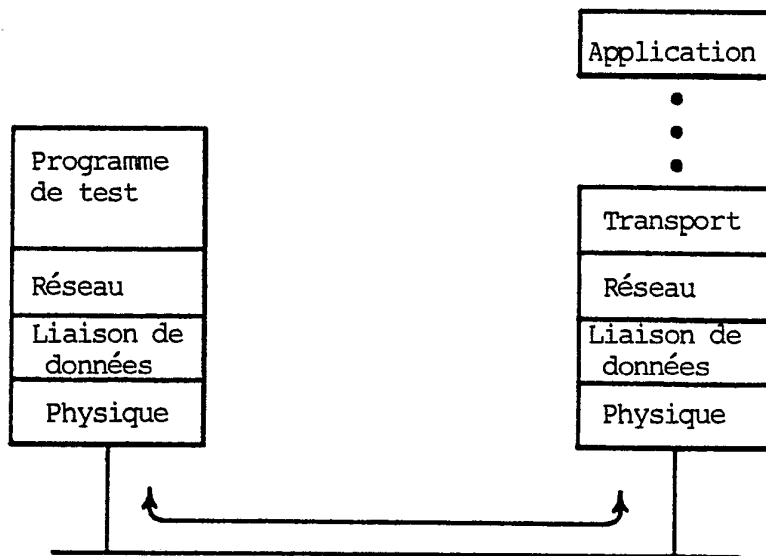


Figure 2.

Données d'utilisateur

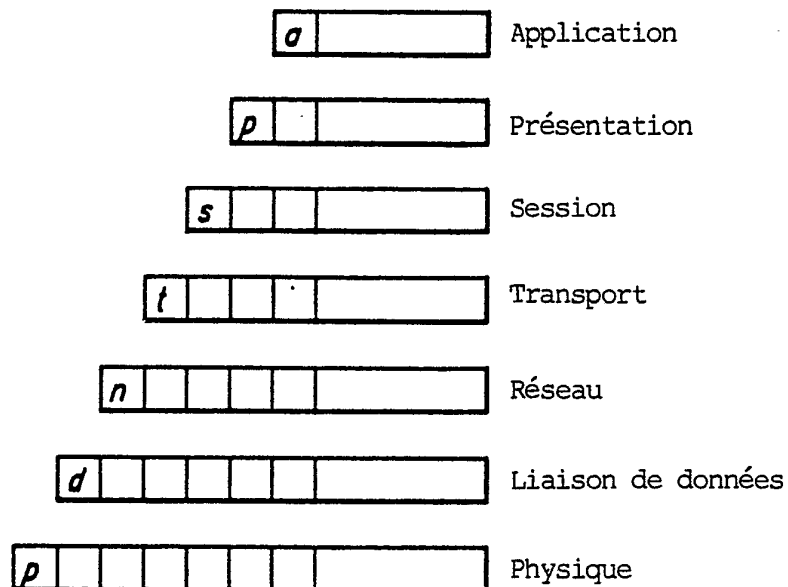
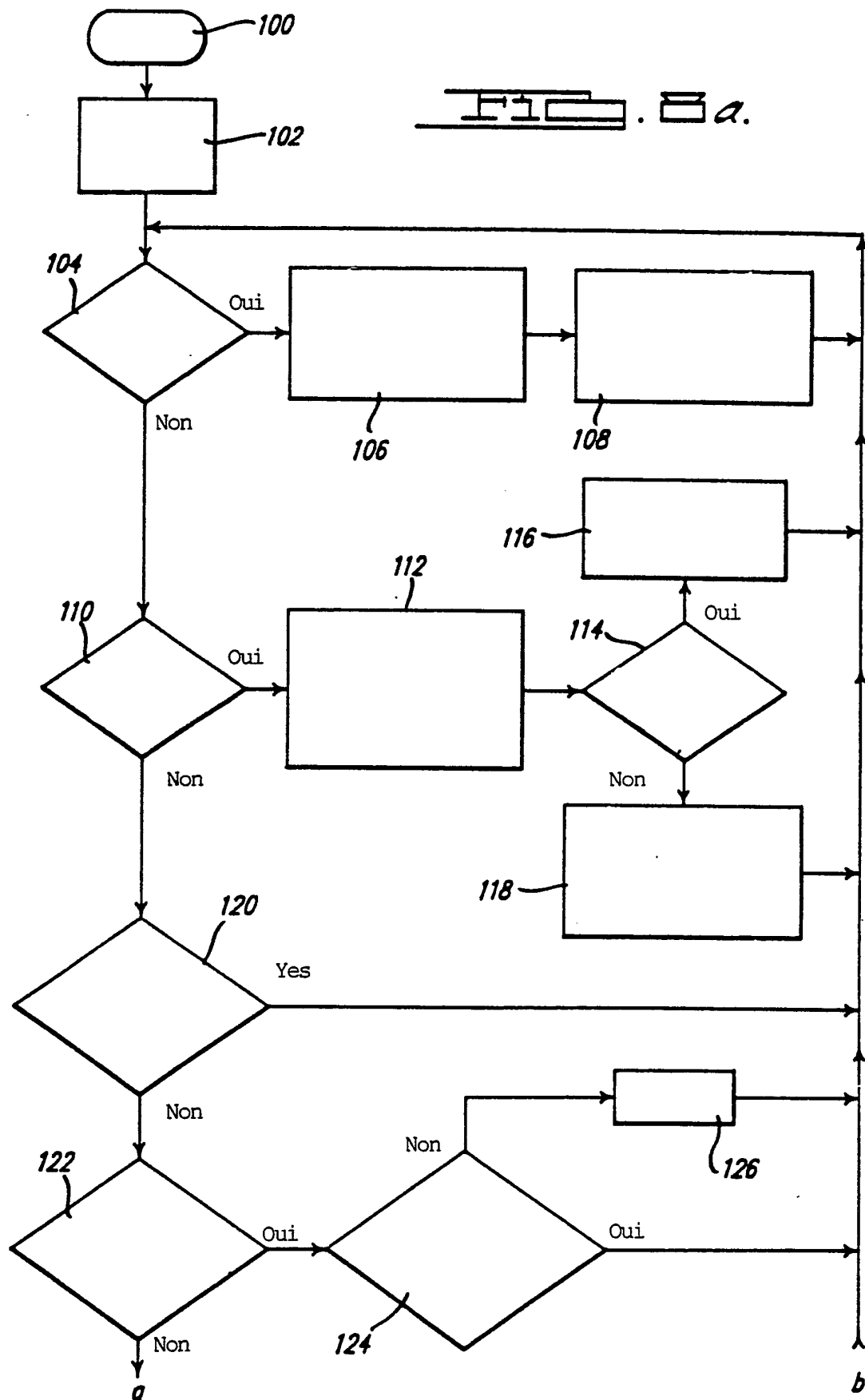
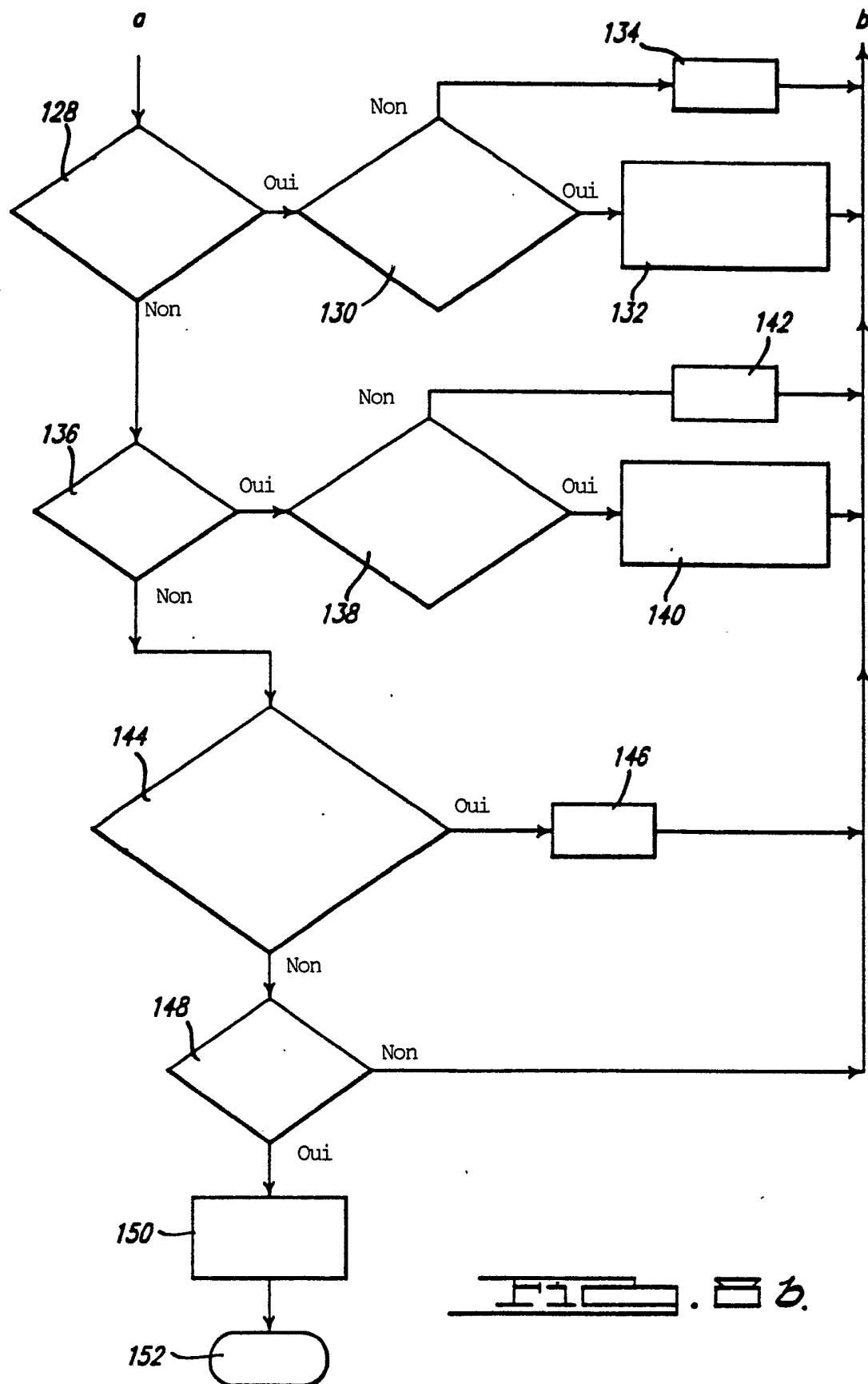


Figure 2.



FIG. 2.



Office européen
des brevets

RAPPORT DE RECHERCHE
établi en vertu de l'article 21 § 1 et 2
de la loi belge sur les brevets d'invention
du 28 mars 1984

Numero de la demande
nationale

BE 8800365
BO 905

DOCUMENTS CONSIDERES COMME PERTINENTS			
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes	Revendication concernée	CLASSEMENT DE LA DEMANDE (Int. Cl.5)
A	IEEE TRANSACTIONS ON COMMUNICATIONS, vol. COM-34, no. 3, mars 1986, pages 257-268, IEEE, New York, US; P.E. GREEN, Jr.: "Protocol conversion" * Page 262, colonne de droite, ligne 28 - page 263, colonne de gauche, ligne 12; page 266, colonne de droite, lignes 26-42 *	1-7	H 04 L 12/00
A	----- GLOBECOM'84, Atlanta, Georgia, 26-29 novembre 1984, vol. 1, pages 157-163, IEEE, New York, US; K. KUROSAWA et al.: "A new specification and validation method for communication protocols - A proposal of a composite state diagram mixed with logical relations" * Page 160, colonne de droite, ligne 30 - page 162, colonne de droite, ligne 4 * -----	1-7	DOMAINES TECHNIQUES RECHERCHES (Int. Cl.5) H 04 L
Date d'achèvement de la recherche 06-11-1991		Examineur DE LA FUENTE DEL AGUA P.	
CATEGORIE DES DOCUMENTS CITES X : particulièrement pertinent à lui seul Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie A : arrière-plan technologique O : divulgation non-écrite P : document intercalaire		T : théorie ou principe à la base de l'invention E : document de brevet antérieur, mais publié à la date de dépôt ou après cette date D : cité dans la demande L : cité pour d'autres raisons & : membre de la même famille, document correspondant	