

①⑨ RÉPUBLIQUE FRANÇAISE
—
**INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE**
—
COURBEVOIE
—

①① **N° de publication :** **3 079 695**
(à n'utiliser que pour les
commandes de reproduction)

②① **N° d'enregistrement national :** **18 52888**

⑤① Int Cl⁸ : **H 04 L 1/00 (2018.01), H 04 L 29/06**

①②

BREVET D'INVENTION

B1

⑤④ **ANALYSE ET FILTRAGE DE DONNEES DANS UN SYSTEME DE TRANSMISSION DE DONNEES.**

②② **Date de dépôt :** 03.04.18.

③⑦ **Priorité :**

④③ **Date de mise à la disposition du public
de la demande :** 04.10.19 Bulletin 19/40.

④⑤ **Date de la mise à disposition du public du
brevet d'invention :** 18.11.22 Bulletin 22/46.

⑤⑥ **Liste des documents cités dans le rapport de
recherche :**

Se reporter à la fin du présent fascicule

⑥⑦ **Références à d'autres documents nationaux
apparentés :**

○ **Demande(s) d'extension :** Polynésie-Fr

⑦① **Demandeur(s) :** *SILKAN RT Société anonyme* — FR.

⑦② **Inventeur(s) :** DELWASSE SERGE et LAPORTE
VINCENT.

⑦③ **Titulaire(s) :** *SILKAN RT Société anonyme.*

⑦④ **Mandataire(s) :** AUGUST DEBOUZY.

FR 3 079 695 - B1



TITRE DE L'INVENTION

Analyse et filtrage de données dans un système de transmission de données.

5 **DOMAINE DE L'INVENTION**

La présente invention concerne le domaine des télécommunications.

Plus particulièrement, la présente invention concerne le transport de données de manière fiable et déterministe.

10 **ARRIERE-PLAN TECHNOLOGIQUE**

Dans un réseau de télécommunications, le transport de données se fait de manière habituelle sous forme de trames de données établies selon un protocole prédéfini. Pour transporter une trame entre un émetteur de données (ou producteur) et un récepteur (ou consommateur), on utilise habituellement un commutateur réalisé à partir d'unités de traitement contrôlées par un logiciel. Un exemple classique de commutateur réseau est le commutateur Ethernet.

Ce type de commutateur ne permet pas de garantir de façon sûre ni la délivrance effective de la trame de données ni un délai maximal de transmission de cette trame. L'utilisation de tels commutateurs dans les systèmes dits « critiques » ou la fiabilité totale des transmissions est nécessaire n'est donc pas possible. En outre, ce type de commutateur est généralement limité à une utilisation selon un protocole de communication physique et logique unique.

Ainsi, il existe un besoin pour un système de communication de données permettant la communication de données de manière intègre et dans un délai maximal garanti, sans limitation d'utilisation d'un protocole particulier.

La présente invention s'inscrit dans ce contexte.

RESUME DE L'INVENTION

Selon un aspect de l'invention, il est prévu un module d'analyse et de filtrage de données pour un système de transmission de données comportant :

- une unité de réception de trame de données à retransmettre,
- une unité de mémoire pour stocker lesdites données de ladite trame au fur et à mesure de leur réception par l'unité de réception,

- au moins une unité de traitement pour réaliser un traitement sur au moins une donnée stockée dans ladite unité de mémoire, le résultat dudit traitement étant destiné à être utilisé dans la gestion de la retransmission des données, et

5 - une unité de commande pour déclencher le chargement de données depuis au moins l'un parmi ladite unité de réception et de l'unité de mémoire vers ladite au moins une unité de traitement, ledit chargement et ledit traitement étant réalisés pendant la réception de ladite trame au fur et à mesure du stockage des données dans ladite mémoire.

10 Un module selon le premier aspect permet une parallélisation maximale de l'arrivée d'une trame, de son décodage et de son filtrage ce qui augmente les performances (vitesse de traitement).

Un module selon le premier aspect permet une compatibilité multi-protocoles.

15 Il peut aussi permettre une augmentation de la sécurité intrinsèque du traitement de données car son implémentation entièrement matérielle ne permet pas de modification de son comportement et que ses capacités à fonctionner plus rapidement que l'arrivée de toute donnée assure sa disponibilité permanente.

Selon des modes de réalisation le fonctionnement du module est synchrone.

Par exemple, l'unité de commande est une machine à états finie.

20 Par exemple encore, l'unité de mémoire comporte un ensemble de registres.

Selon des modes de réalisation, le chargement de données est déclenché en fonction de la disponibilité desdites données dans un registre donné.

Selon des modes de réalisation, le module comporte une unité de traitement permettant le contrôle d'intégrité de ladite trame en cours de réception.

25 Par exemple, le contrôle d'intégrité porte sur une donnée chargée depuis ladite unité de réception.

Selon des modes de réalisation, le module comporte une unité de traitement qui permet le contrôle de la validité de ladite trame en cours de réception.

30 Par exemple, l'unité de traitement permettant le contrôle de la validité de ladite trame en cours de réception opère un traitement dichotomique de ladite trame.

Selon des modes de réalisation, l'unité de traitement permettant le contrôle de la validité de ladite trame en cours de réception comporte une pluralité d'étages en série.

35 Selon des modes de réalisation, au moins un étage de l'unité de traitement comporte une pluralité de sous-étages parallèles.

Par exemple, le module comporte une unité de traitement (503) permettant la réalisation de calculs spécifiques pour alimenter au moins l'un parmi un étage et un sous-étage de ladite unité de traitement permettant le contrôle de la validité de ladite trame en cours de réception.

Selon des modes de réalisation, l'unité de commande permet en outre de commander l'unité de réception en fonction d'information reçue de ladite au moins une unité de traitement.

Par exemple, le module est configuré pour générer une sortie indiquant une anomalie de réception de ladite trame en fonction d'information reçue de ladite au moins une unité de traitement.

Par exemple encore, le module est configuré pour générer une sortie indiquant le contenu de ladite unité de mémoire en fonction d'information reçue de ladite au moins une unité de traitement.

Selon des modes de réalisation, le module est configuré pour générer un pointeur en sortie par au moins une unité de traitement pointant vers un espace mémoire du système contenant une structure de données pour encapsuler ladite trame de données à retransmettre.

Selon des modes de réalisation, le module est réalisé dans un circuit programmable ou un circuit intégré propre à une application (ASIC).

BREVE DESCRIPTION DES FIGURES

D'autres caractéristiques et avantages de l'invention apparaîtront à la lecture de la présente description détaillée qui suit, à titre d'exemple non limitatif, et des figures annexées parmi lesquelles:

- la **figure 1** illustre un système selon des modes de réalisation;
- la **figure 2** illustre la transmission d'une donnée selon des modes de réalisation,
- les **figures 3a-3b** illustrent la réception d'une donnée (ou message) par un module d'échange selon des modes de réalisation,
- la **figure 4** illustre le traitement d'une trame conforme au standard IEEE 802.3 selon des modes de réalisation,
- la **figure 5** est un schéma illustrant un module de filtrage selon des modes de réalisation.

35

DESCRIPTION DETAILLEE DE L'INVENTION

Un système selon des modes de réalisation est décrit en référence à la **figure 1**.

Le système représenté sur cette figure est réalisé de manière matérielle, c'est-à-dire qu'il n'est pas contrôlé de manière logicielle. Le système est par exemple réalisé sous forme de circuit de type FPGA (acronyme de « Field-Programmable Gate Array »).

Le système comporte un ou plusieurs modules d'échange de données **100** (ComSet) et un ou plusieurs modules de base de données **101** (Station de base).

Chaque base de données permet le stockage de données échangées entre les modules d'échange. Le nombre de modules d'échange est égal au nombre de ports de communication souhaités pour le système. Ces ports de communication sont connectés

Les modules d'échange **100** peuvent être connectés à une base de données **101** au moyen d'une interface standardisée, c'est-à-dire que l'interface est la même pour tous les modules de communication. L'interface standard comporte autant de bus de transport de données que de modules en communication directe avec la base de données.

Par exemple, un module **112** est en charge d'encapsuler des trames de données reçues et à traiter au sein du système. Ce module communique avec la base de données via un bus dédié **102**. Un module **113** est par exemple en charge de la génération de commandes à destination de la base de données pour extraire une donnée à retransmettre. Ce module communique avec la base de données via un bus dédié **103**. Un module **115** est en charge de recevoir des trames encapsulées de la part de la base de données et de les traiter pour une retransmission. Ce module communique avec la base de données via un bus dédié **105**. Un module **114** est en charge de gérer les émissions de la part de la base de données indiquant certains évènements tel que le stockage de trames encapsulées. Cette gestion inclut par exemple la coordination des modules **113** et **115**. Ce module communique avec la base de données via un bus dédié **104**. Un module d'horloge **116** est en charge de de réceptionner les mises à jour de l'heure générale du système depuis la station de base. Par ailleurs, il fournit l'heure à tous les modules du système de transmission de données qui doivent soit marquer les données, soit vérifier les délais de transmission de celles-ci. Le module d'horloge peut aussi générer les évènements pour la fourniture des signaux de synchronisation au processus qui en ont fait la demande. Il peut en outre prévenir les modules d'émission

de données du moment de l'envoi de données lorsqu'elles sont définies comme cycliques par le paramétrage. Le module d'horloge communique avec la base de données via un bus dédié **106**.

5 Les bus de communication sont par exemple basés sur des mémoires à double accès qui permettent une indépendance de fonctionnement des modules amonts et aval. Ils permettent une latence minimale et ont une bande passante très supérieure au débit maximal de chaque transmission et peuvent donc fonctionner en mode famine.

10 Les données échangées par un module d'échange sont reçues et envoyées au travers d'une interface **107** configurée en fonction du protocole utilisé à l'extérieur du système. Des exemples de protocoles avec lesquels le système peut être interfacé sont Ethernet, CAN, ARINC 664P7 ou autres. Le module **107** peut communiquer avec les autres modules du système via des interfaces asynchrones dédiées.

15 La communication entre les différents sous-modules **112, 113, 114, 115** de chaque module d'échange avec les interfaces avec les bus de communication respectifs **102, 103, 104, 105** se fait via des dispositifs **108, 109, 110, 111** permettant de garantir l'indépendance des sous-modules en termes de fréquence d'accès et d'espace de stockage. Par exemple, il s'agit de dispositifs de type DAM (acronyme de « Dual Access Memory »).

20 Le module **117** est un module de traitement des trames de données reçues avant leur encapsulation par le module **112**. Il est décrit plus en détails dans la suite.

A l'inverse, le module **118** permet de des-encapsuler une trame encapsulée reçue du module **115**.

25 Un module de gestion de statut **119** est en charge de garder les états de fonctionnement courant de chacun des modules du système de transmission de données si le paramétrage le demande. De plus tous les modules sont conçus pour transmettre tout incident de fonctionnement à ce module de gestion de statut qui met alors à jour sa base d'information et génère un évènement d'erreur immédiatement si le paramétrage le demande vers le système de transmission de données destiné à cet effet.

30 En référence à la **figure 2**, on décrit la transmission d'une donnée (ou message) par un module d'échange **200**.

35 Le message **201** est reçu au niveau du module d'interface **202**. Ce module d'interface est en charge du dialogue avec les composants responsables de la gestion physique des signaux du protocole lié au système de transmission de données. Il récupère l'information des trames reçues via ces composants et les transmet octet par

octet au module suivant. Le message **201** est ensuite transmis au module **203** en charge de l'analyse et du filtrage du message (fonctionnement décrit plus en détail dans ce qui suit).

5 Le module **203** vérifie l'intégrité du message et vérifie aussi que le message est autorisé, par configuration, à être reçu par le module d'échange **200**. En d'autres termes, le module **203** extrait les informations de la trame entrante afin de s'assurer que l'adresse IP source et le port source du message ainsi que sa taille sont bien tels que le paramétrage les a définis. D'autre part le calcul du CRC permet de vérifier que la trame reçue est bien intègre.

10 Le message est ensuite encapsulé dans un module d'encapsulation **204**. Une fois encapsulé, le message est traité indépendamment du protocole utilisé pour le transporté à l'extérieur du système. Le message encapsulé est ensuite transmis à la base de données **205**, via le bus **206**, pour stockage en mémoire.

15 Le message encapsulé peut être stocké en prenant en compte différents paramètres. Il s'agit par exemple d'éviter la duplication d'information en mémoire. L'enregistrement d'un message est unique même si il possède plusieurs destinataires. Chaque destinataire est prévenu de l'arrivée de la donnée ainsi que de son emplacement. Les destinataires sont alors autorisés à venir l'y chercher pour transmission. L'heure d'arrivée de la donnée fait aussi partie des données d'encapsulation afin que les
20 récepteurs de l'information puissent contrôler la validité temporelle de celle-ci. Il s'agit par exemple encore de tenir compte du vieillissement du message. Il s'agit par exemple encore de gérer les méthodes d'accès au message qui définissent les adresses des objets en mémoire (on peut par exemple citer les méthodes dites de « *Sampling* », « *Queuing* », « *Flip/Flop* » qui sont des méthodes standard.

25 En référence aux **figures 3a-3b**, on décrit la réception d'une donnée (ou message) par un module d'échange **300**.

A la suite du stockage du message encapsulé dans la base de données **205** décrite ci-dessus en référence à la **figure 2**, une donnée de type « événement » est envoyée au module **300** à qui le message **201** est destiné via le bus **304**. Une donnée
30 de type « événement ». Une donnée de type « événement » est un objet au format d'encapsulation commun dont l'objet et de prévenir le module destinataire soit du comportement d'un autre module soit d'une action à accomplir pour ce dernier.

35 La base de données **205** (commune aux modules **200** et **300**) génère et gère les événements relatifs à l'émission d'un message encapsulé. La base de données cadence aussi les moments de transmission des données de type « événement » aux

modules d'échange qui doivent émettre le message encapsulé concerné par la donnée. Le module d'émission est prévenu par message que la donnée à émettre est disponible et où aller la chercher.

La donnée de type « évènement » est ensuite reçue par un module **301** de gestion des émissions cycliques et évènementielles en charge de la réception de ces données. Sur réception d'une telle donnée le module **301** envoie une demande de commande à destination du module de génération de commandes **302**.

Le module de génération de commandes **302** envoie ensuite une commande à destination de la base de données afin que l'objet encapsulé soit envoyé vers le module d'échange **300**.

En parallèle, le module **301** envoie une demande de mise en mode attente du message encapsulé au module **303** de génération de données à émettre.

Le module **302** joue ainsi un rôle de gestion du cycle (en particulier de la fréquence) d'envoi des messages.

La base de données envoie ensuite le message encapsulé via le bus **305** au module **303** qui désencapsule le message pour en faire un message conforme au protocole de l'équipement connecté sur le port physique du module d'échange **300**.

Le message ainsi désencapsulé est transmis via un module **306** de contrôle de l'émission et le module d'interface **307**. Ce module de contrôle après une dernière vérification de l'intégrité de la donnée à l'aide du code de contrôle (CRC) vérifie aussi que les informations contenues correspondent bien à tous les critères définis par le paramétrage du port de sortie.

Dans ce qui suit, on décrit les opérations réalisées par le module d'analyse et de filtrage d'un module d'un module d'échange.

Le filtrage intervient entre la réception d'une trame et son encapsulation. Le filtrage a pour objectif de vérifier que la trame reçue est bien une trame connue, attendue et bien formée. Si ce n'est pas le cas, la trame est rejetée et un message d'erreur est généré. Le filtrage a aussi pour objectif de fournir, si la trame est acceptée, l'information des emplacements où les données nécessaires à l'encapsulation de la trame peuvent être récupérées.

D'une manière générale, le filtrage, dans le domaine des réseaux, est l'opération qui consiste à vérifier l'intégrité et la validité d'une trame en réception.

La vérification de l'intégrité de la trame permet de s'assurer qu'une trame reçue respecte des propriétés liées à sa structure.

Parmi les vérifications effectuées on trouve par exemple les altérations au niveau logique élémentaire (bit ou ensemble de bits). La vérification est réalisée grâce à un code détecteur d'erreurs du type contrôle de redondance cyclique ou CRC (acronyme de « *Cyclic Redundancy Check* »).

5 On trouve aussi parmi les vérifications effectuées le fait que l'arrivée de deux trames successives doit respecter un délai minimum (on parle d'IFG, acronyme de « *InterFrame Gap* »). Si ce délai n'est pas respecté, la trame est considérée corrompue.

D'autres vérifications sont possibles comme par exemple les intervalles de longueur de la trame, notés [LMAX, LMIN] qui peuvent dépendre des protocoles. Une
10 trame peut avoir une taille fixe (c'est par exemple le cas dans le protocole Arinc 429) ou une taille variable bornée en valeurs inférieure et supérieure (par exemple le protocole Ethernet 802.3). En plus des restrictions normatives, l'utilisateur peut durcir les conditions sur les tailles. Quand la trame ne respecte pas les contraintes de taille, elle est considérée corrompue.

15 La vérification de la validité permet de s'assurer qu'une trame reçue à un point de réception donné est bien autorisée à être reçue à cet endroit (le vocable « *endroit* » sous-entend « *port de communication* » dans la terminologie réseau). La prise de décision de l'autorisation se base sur les données d'encapsulation de la trame. Si on prend l'exemple des trames de type IP (acronyme de « *Internet Protocol* »),
20 respectant le standard IEEE 802.3, les données d'encapsulation sont à récupérer des champs de la pile protocoles tels que : « *IP sources* », « *EtherType* », « *MAC source* », « *MAC destination* », etc.

Chaque trame qui passe par le module de filtrage provoque une prise de décision binaire, donc avec deux états stables et aucune indécision possible : la trame
25 passe le filtrage ou la trame ne passe pas le filtrage. Toute violation d'intégrité et/ou de validité de la trame provoque son rejet (trame ne passe donc pas le filtrage).

Deux actions sont alors déclenchées. Une première action est la suppression (purge) de la trame du module de filtrage. Une deuxième action est une levée d'un statut d'erreur par le contrôle du module de gestion du statu évoqué ci-avant. Les statuts
30 peuvent être gérés de différentes manières. Par exemple, il est possible de choisir l'envoi d'une information d'état de façon cyclique et ce, avec une périodicité paramétrable, ou d'émettre un évènement dès la survenance d'un évènement particulier vers le module et le système de transmission de données paramétrés à cet effet. Les statuts peuvent être actifs et donc pris en compte ou masqués et donc ignorés.

Dans le cas où aucune violation des règles d'intégrité et de validité n'est détectée, la trame est acceptée dans le système, autrement dit, la trame passe le filtrage.

Deux actions sont alors déclenchées. Une première action est la fourniture de la trame au module suivant (c'est-à-dire le module d'encapsulation). Une deuxième action est la génération d'un pointeur vers les tables d'encapsulation qui permet de trouver, dans les mémoires internes, les données de certains champs d'encapsulation objet (sur-encapsulation de la trame).

L'encapsulation d'une trame est en fait une sur-encapsulation. Une trame est déjà une encapsulation des données utiles par des champs propres au protocole. Ici, il s'agit d'ajouter d'autres champs. Il est alors possible de parler de sur-encapsulation. Certains champs sont calculés et chargés en mémoire par un configurateur en mode hors-ligne (avant le démarrage de la phase opérationnelle) et d'autres champs sont calculés par le composant en mode en-ligne (pendant la phase opérationnelle).

Une particularité du module de filtrage, outre le fait qu'il soit conçu pour une implémentation exclusivement matérielle, est la garantie d'une performance très élevée comparativement aux systèmes existants sur le marché. L'aspect « implémentation matérielle » ici fait opposition à une mise en œuvre logicielle basée sur un schéma type Von Neumann. Le module de filtrage, comme le module d'échange de données auquel il appartient est implantable dans des circuits programmables type FPGA (acronyme de « *Field-Programmable Gate Array* » ou ASIC (acronyme de « *application-specific integrated circuit* »). La performance allie à la fois débit élevé et déterminisme temporel.

Le fonctionnement du module de filtrage comporte une base de registres de réception des trames, des modules de dichotomie, et une machine d'état de contrôle.

Tout le mécanisme du module de filtrage est cadencé par une machine d'état de contrôle qui pousse dans une structure de registres la trame au fur et à mesure qu'elle est reçue, et, à des moments opportuns, charge des données particulière et/ou fait des calculs sur certaines de ces données.

Avant même la fin de la réception de la trame, le filtrage, qui utilise des tables de dichotomie pour la recherche des informations et actions à effectuer, est déjà lancé car toutes les informations nécessaires sont rendues disponibles.

Au besoin, il est possible d'arrêter l'opération, et de purger le module si, par exemple, une corruption de donnée est détectée via les mécanismes de protection (tels que le CRC (acronyme de « *Cyclic redundancy check* » ou les sommes de contrôle « *checksum* » en anglais).

Dans ce fonctionnement, plusieurs traitements sont parallélisés. En effet, quand l'arrivée d'une trame est détectée, celle-ci est dé-sérialisée au fil de son arrivée. A chaque fois qu'un nombre suffisant de bits est correctement reçu, la machine d'état remplit un registre de réception dans la base de registres prévue à cet effet. Chaque
5 registre est clairement identifié. Ainsi, en fonction du protocole de la trame en cours de réception, il est possible de récupérer à la volée des informations particulières pour les besoins, soit de traitement, soit de passage de paramètres pour une fonction donnée.

L'exemple d'une trame conforme au standard IEEE 802.3 est décrit en référence à la **figure 4**.

10 Dans l'exemple d'une telle trame, les 6 premiers octets reçus sont l'adresse MAC destination, les 6 octets suivants sont l'adresse MAC source, les 2 octets suivants sont le champ EtherType. Si pour les besoins du filtrage, le module nécessite les deux derniers octets de l'adresse MAC de destination (numéro du « *Virtual Link* » dans la norme Arinc 664P7), et si la base de registre est constituée de registres de 32 bits (4
15 octets) alors la réception du second registre, noté « **Reg2** » dans la figure 4, déclenche une récupération des octets 1 et 2 pour les besoins du traitement (identification du « *Virtual Link* »).

L'opération de réception et de stockage, en plus des tâches de récupération des paramètres pendant la réception, assure la vérification de l'intégrité des trames :
20 vérification de l'enveloppe de la trame, de l'IFG, de la taille, du CRC, etc.

Les informations récupérées de l'encapsulation (parfois après un traitement dessus) sont utilisées pour s'assurer de la validité de la trame. En d'autres termes, vérifier si la trame est bien autorisée à passer dans la suite du système. Cette activité est prise en charge par les « tables de dichotomie » appelées aussi « tables de filtrage ».

25 Pour une optimisation des performances, le schéma de la dichotomie est divisé en plusieurs étages dont chaque étage peut être lui-même subdivisé en plusieurs parties. Le principe de partitionnement en « étages » et « parties » obéit aux deux règles suivantes. Les « étages » sont « pipelinés », c'est-à-dire, les sorties d'un « étage » d'ordre n-1 sont des entrées pour l'étage d'ordre n. L'exécution de toutes les « parties »
30 dans un même « étage » est parallélisée.

Dans un exemple de mise en œuvre, la dichotomie est organisée en deux étages dont le premier est lui-même subdivisé en deux parties. Chronologiquement, les recherches du premier étage sont parallélisées. Les sorties du premier étages constituent des entrées pour le second étage qui, en sortie, fournit un pointeur vers des

structures mémoires qui permettent au module d'encapsulation de sur-encapsuler de la trame reçue.

Le **figure 5** est un schéma illustrant un module de filtrage selon des modes de réalisation.

5 Le fonctionnement global de ce module est synchrone. Ce module utilise une horloge unique distribuée. Il est en outre géré par une machine à états finie **501** (FSM acronyme de « *Final State Machine* »). L'intérêt de cette manière de faire est de pouvoir maîtriser le déclenchement et la durée de chaque opération ce qui induit une garantie d'un temps de traitement borné et connu, et donc, une garantie de déterminisme
10 temporel.

La trame en réception est dé-sérialisée dans le module d'entrée **500** « Data_IN » qui la stocke au fur et à mesure de son arrivée dans une base de registres **507**. Dès la disponibilité d'informations pertinentes dans cette dernière, celles-ci sont récupérées pour être exploitées. Dans l'exemple de la figure 5, le premier octet du
15 premier registre peut être mis à disposition, dès sa réception, du module **502** pour la recherche dichotomique de l'étage 1 partie 2, notée E1P2. Le lancement du bloc **502** nécessite une commande de la machine **501** ainsi que des données complémentaires (notées « data externes » telle que l'enveloppe de trame et l'horloge).

En parallèle, la réception de la trame continue et d'autres informations sont
20 prélevées à la volée soit pour alimenter les recherches dichotomiques, soit pour alimenter une unité **503** de calcul spécifiques qui alimente par la suite les recherches dichotomiques. L'unité **503** permet de traiter certaines informations d'encapsulation qui ne sont pas utilisables dans leur état brut et/ou doivent être combinées entre elles pour qu'elles soient exploitables par la suite. Par exemple, le module **502** ne peut être lancé
25 qu'après extraction directe de l'octet 2 du registre 2 et mise à disposition d'un résultat de traitement de l'unité **503** basé sur l'octet 1 du registre 2. Le lancement du module **504** comprenant l'étage 2 de la dichotomie, noté E2, ne peut se faire qu'une fois les traitements de l'étage 1 complétés. En effet, les résultats des modules **502** (E1P1) et **505** (la partie 1 de l'étage 1 notée E1P2) sont injectés comme entrées dans le module
30 **504**. De plus, ces entrées ne sont pas exclusives. Chaque étage peut avoir besoin d'entrées complémentaires (en plus de celle de l'étage amont). Dans l'exemple de la figure 5, le module **504** nécessite le dernier octet de l'avant-dernier registre ainsi qu'un résultat fourni par l'unité **503**.

En parallèle de tous ces traitements, une unité **506** de contrôle d'intégrité
35 vérifie que la trame en réception ne comporte pas d'anomalie. Les vérifications portent

sur plusieurs paramètres tel que par exemple le CRC, la longueur, l'IFG, ou autre. Cette unité informe la machine **501** du statut de la réception en cours et lève une anomalie le cas échéant.

5 Tout au long du fonctionnement, des statuts sont générés et échangés afin de prévenir tout dysfonctionnement et détecter toute anomalie.

10 Les sorties du modules de filtrage peuvent être au nombre de un ou trois selon l'état final de la machine d'état de contrôle. Une sortie est un mot de statut contenant l'état final de la machine d'état avec, le cas échéant, une indication (communément appelée drapeau ou « *flag* » *en anglais*) à propos de l'anomalie constatée. Si le traitement s'est correctement déroulé, alors le module de filtrage fournit en plus le contenu de la base de registres qui n'est autre que la trame reçue et un pointeur, issu du dernier étage de la dichotomie, sur les structures mémoires permettant de créer l'objet issu de l'encapsule de la trame.

15 La présente invention a été décrite et illustrée dans la présente description détaillée en référence aux figures jointes. Toutefois la présente invention ne se limite pas aux formes de réalisation présentées. D'autres variantes, modes de réalisation et combinaisons de caractéristiques peuvent être déduits et mis en œuvre par la personne du métier à la lecture de la présente description et des figures annexées.

20 Pour satisfaire des besoins spécifiques, une personne compétente dans le domaine de l'invention pourra appliquer des modifications ou adaptations.

25 Dans les revendications, le terme "comporter" n'exclut pas d'autres éléments ou d'autres étapes. L'article indéfini « un » n'exclut pas le pluriel. Une ou plusieurs autres unités peuvent être utilisées pour mettre en œuvre l'invention. Les différentes caractéristiques présentées et/ou revendiquées peuvent être avantageusement combinées. Leur présence dans la description ou dans des revendications dépendantes différentes, n'exclut pas en effet la possibilité de les combiner. Les signes de référence ne sauraient être compris comme limitant la portée de l'invention.

REVENDEICATIONS

1. Module d'analyse et de filtrage de données (117, 203) pour un système de transmission de données comportant :

- 5 - une unité de réception (500) de trame de données à retransmettre,
 - une unité de mémoire (507) pour stocker lesdites données de ladite trame au fur et à mesure de leur réception par l'unité de réception,
 - au moins une unité de traitement (505, 502, 504) pour réaliser un traitement sur au moins une donnée stockée dans ladite unité de mémoire, le résultat dudit
10 traitement étant destiné à être utilisé dans la gestion de la retransmission des données, et
 - une unité de commande (501) pour déclencher le chargement de données depuis au moins l'un parmi ladite unité de réception et de l'unité de mémoire vers ladite au moins une unité de traitement, ledit chargement et ledit traitement étant réalisés
15 pendant la réception de ladite trame au fur et à mesure du stockage des données dans ladite mémoire.

2. Module selon la revendication 1, dans lequel le fonctionnement du module est synchrone.

20

3. Module selon l'une des revendications précédentes, dans lequel ladite unité de commande est une machine à états finie.

4. Module selon l'une des revendications précédentes, dans lequel ladite unité
25 de mémoire comporte un ensemble de registres.

5. Module selon la revendication 4, dans lequel le chargement de données est déclenché en fonction de la disponibilité desdites données dans un registre donné.

30

6. Module selon l'une des revendications précédentes, comportant une unité (506) de traitement permettant le contrôle d'intégrité de ladite trame en cours de réception.

35

7. Module selon la revendication 6, dans lequel le contrôle d'intégrité porte sur une donnée chargée depuis ladite unité de réception.

8. Module selon l'une des revendications précédentes, comportant une unité de traitement qui permet le contrôle de la validité de ladite trame en cours de réception.

5 9. Module selon la revendication 8, dans lequel ladite unité de traitement permettant le contrôle de la validité de ladite trame en cours de réception opère un traitement dichotomique de ladite trame.

10 10. Module selon l'une des revendications 8 à 9, dans lequel ladite unité de traitement permettant le contrôle de la validité de ladite trame en cours de réception comporte une pluralité d'étages en série (505, 502, 504).

11. Module selon la revendication 10, dans lequel au moins un étage de l'unité de traitement comporte une pluralité de sous-étages parallèles (502, 505).

15 12. Module selon l'une des revendications 10 et 11, comportant une unité de traitement (503) permettant la réalisation de calculs spécifiques pour alimenter au moins l'un parmi un étage et un sous-étage de ladite unité de traitement permettant le contrôle de la validité de ladite trame en cours de réception.

20 13. Module selon l'une des revendications précédentes, dans lequel ladite unité de commande (501) permet en outre de commander l'unité de réception (500) en fonction d'information reçue de ladite au moins une unité de traitement (505, 502, 504).

25 14. Module selon l'une des revendications précédentes, configuré pour générer une sortie indiquant une anomalie de réception de ladite trame en fonction d'information reçue de ladite au moins une unité de traitement.

30 15. Module selon l'une des revendications précédentes, configuré pour générer une sortie indiquant le contenu de ladite unité de mémoire en fonction d'information reçue de ladite au moins une unité de traitement.

35 16. Module selon la revendication 15, configuré pour générer un pointeur en sortie par au moins une unité de traitement pointant vers un espace mémoire du système contenant une structure de données pour encapsuler ladite trame de données à retransmettre.

17. Module selon l'une des revendications précédentes réalisé dans un circuit programmable ou un circuit intégré propre à une application (ASIC).

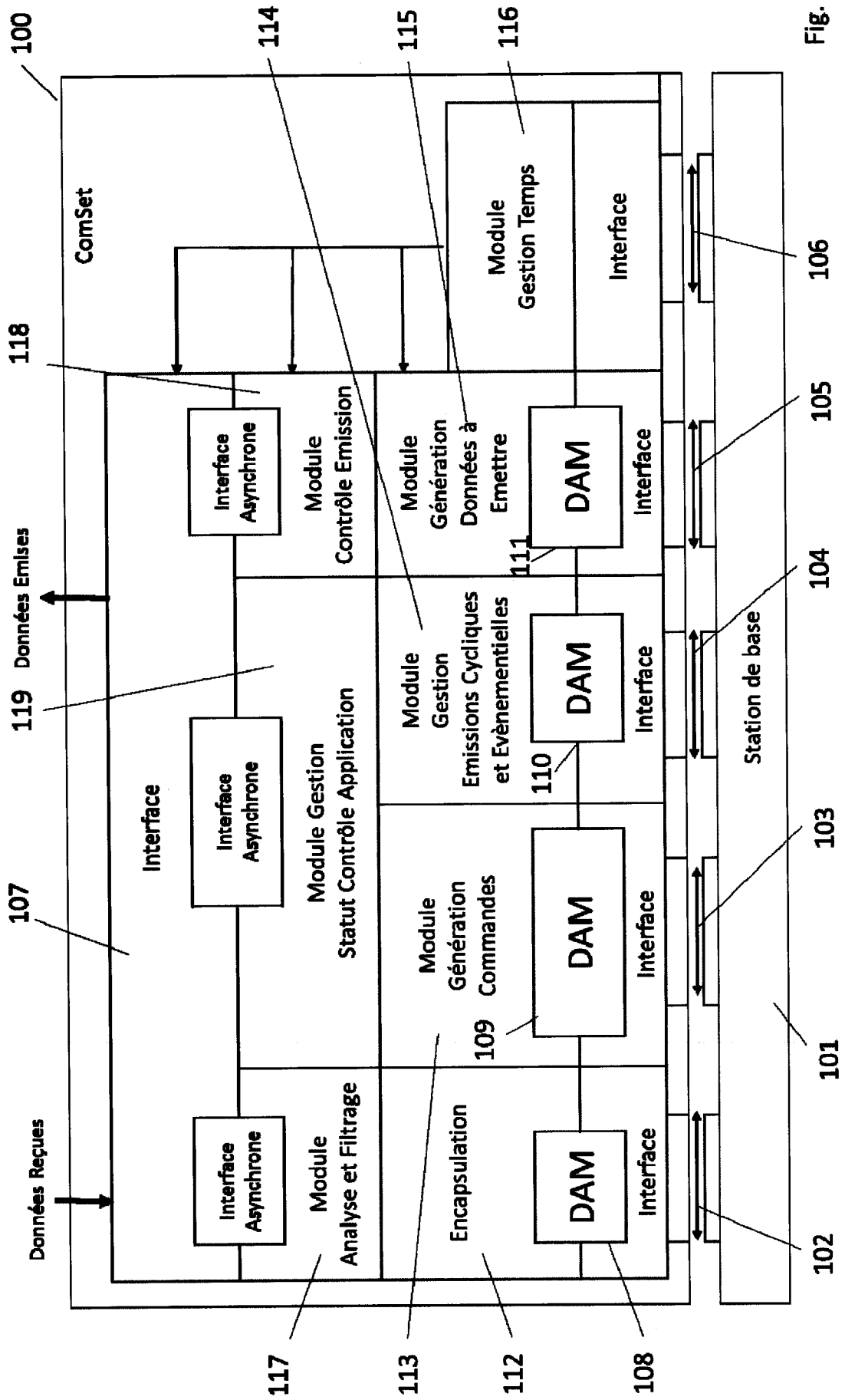


Fig. 1

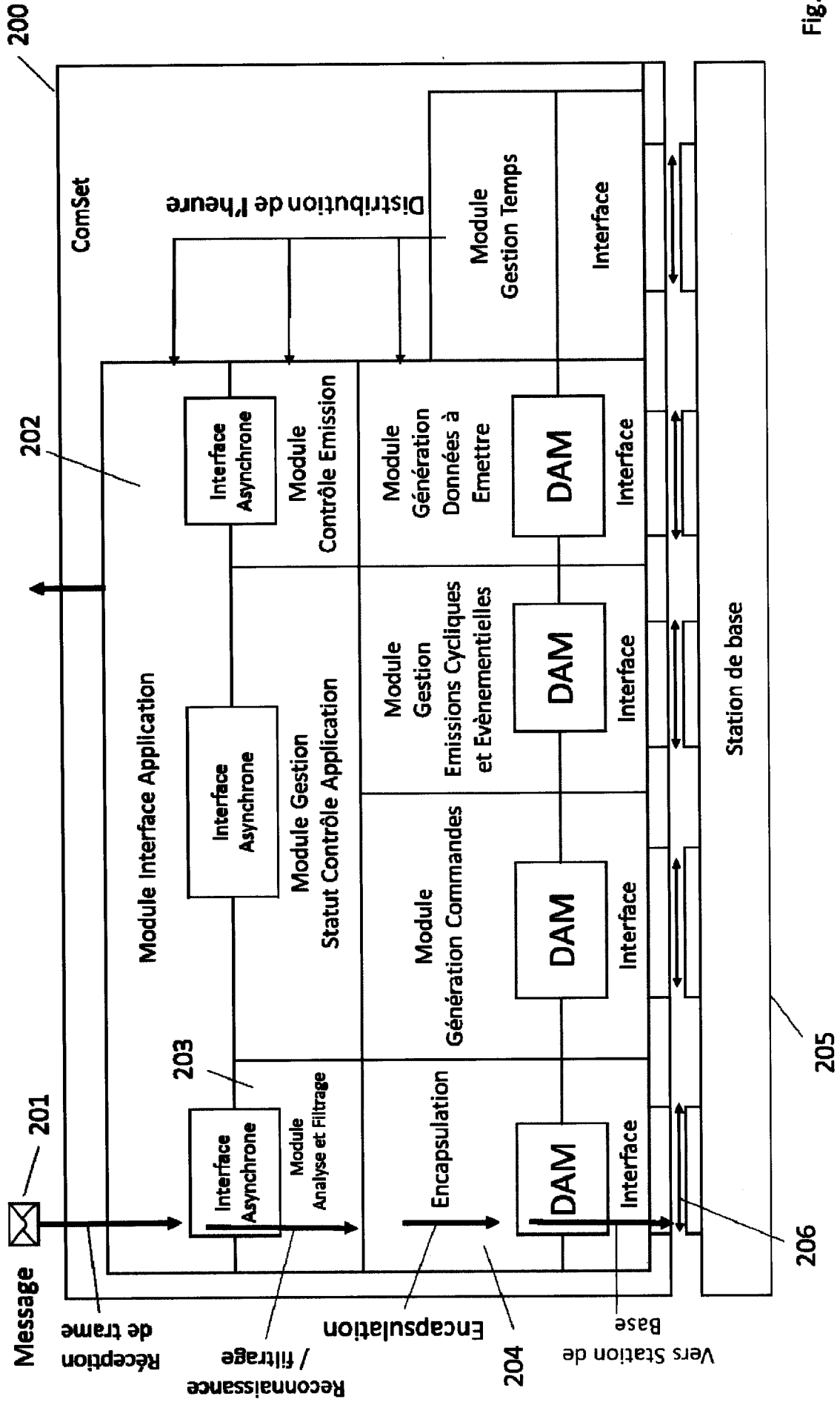


Fig. 2

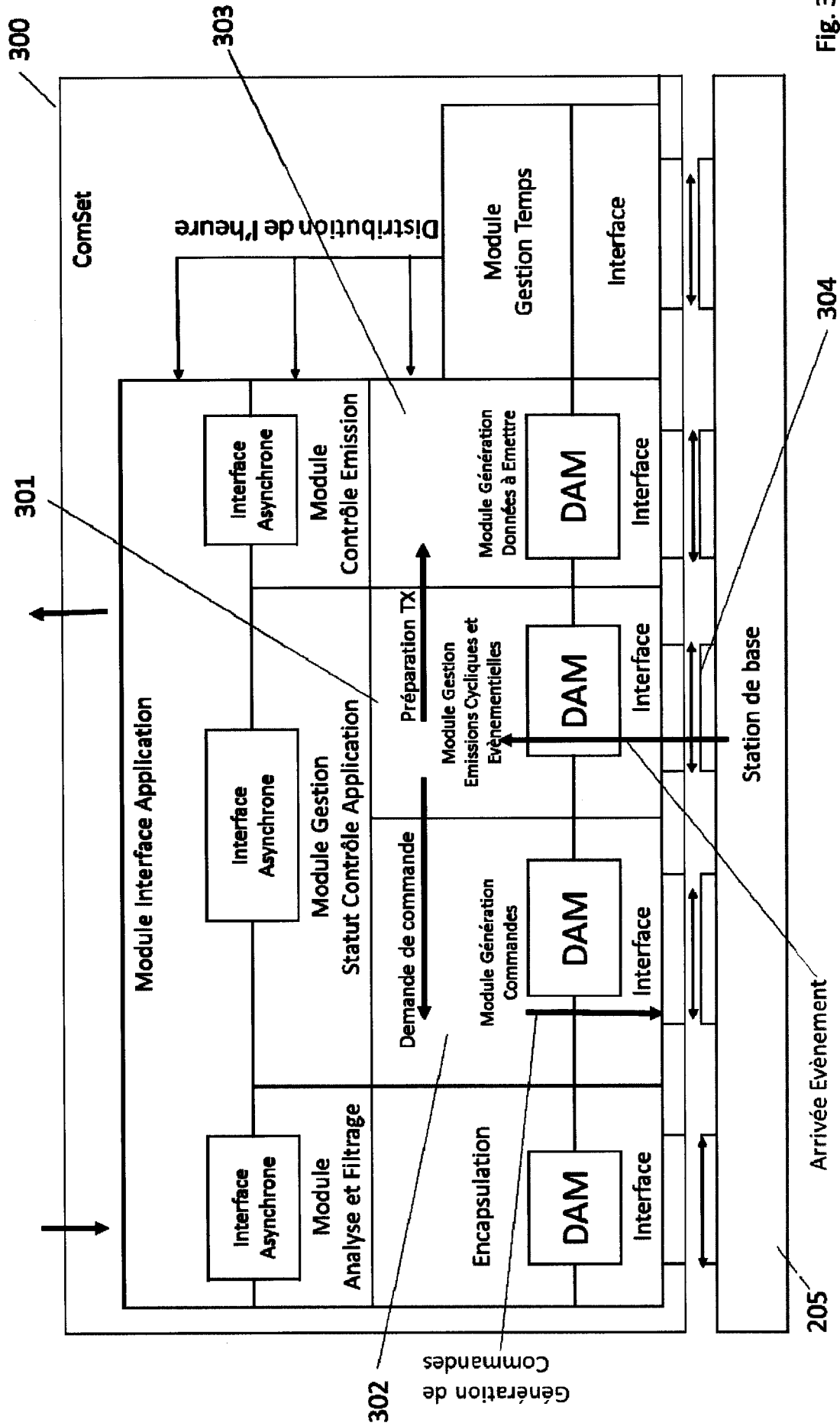


Fig. 3a

1	2	3	4
----------	----------	----------	----------

Reg1

Reg2

Reg3

Reg4

MAC DEST	MAC DEST	MAC DEST	MAC DEST
MAC DEST	MAC DEST	MAC SRC	MAC SRC
MAC SRC	MAC SRC	MAC SRC	MAC SRC
ETH TYP	ETH TYP		

Fig. 4

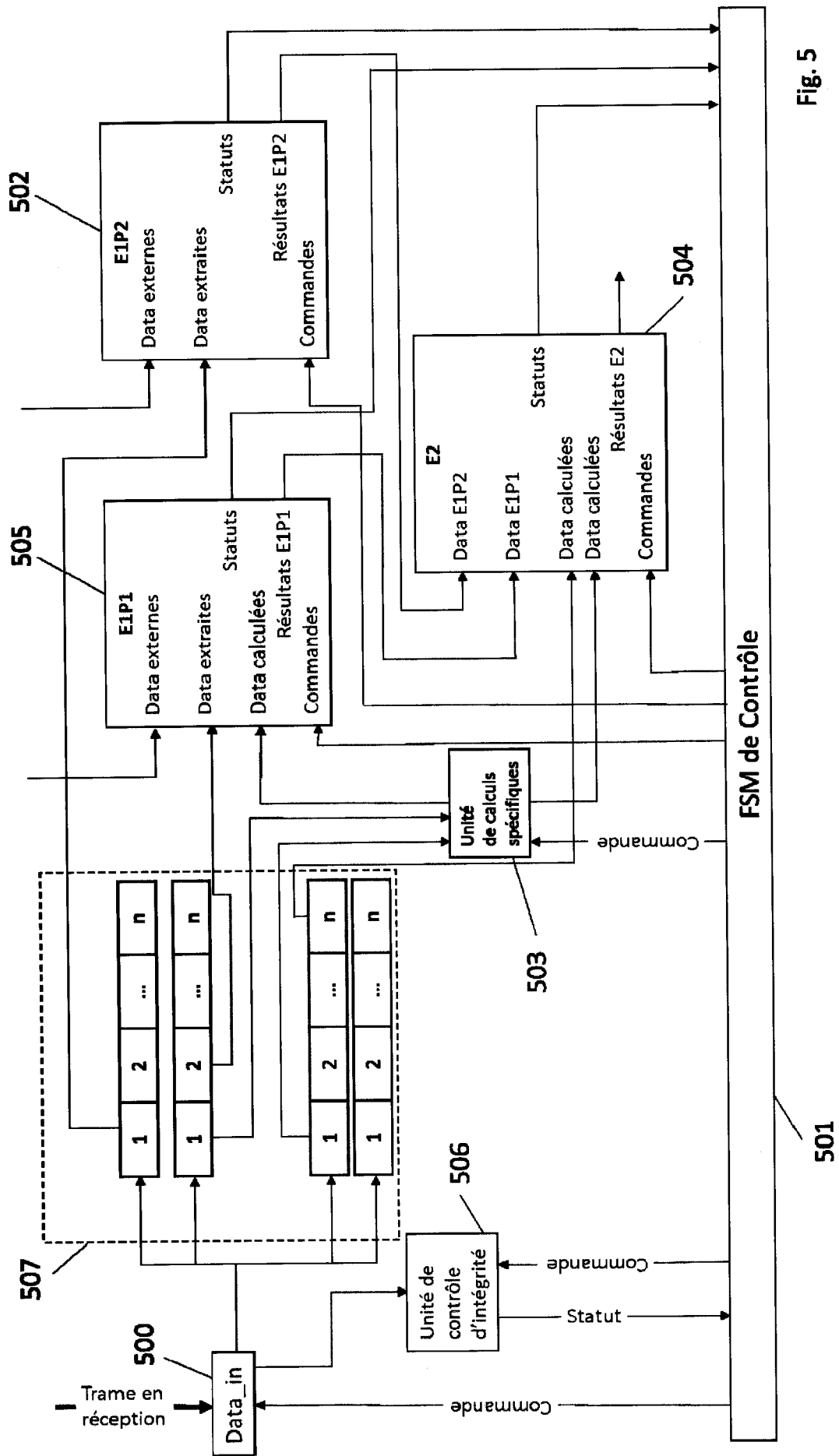


Fig. 5

RAPPORT DE RECHERCHE

articles L.612-14, L.612-53 à 69 du code de la propriété intellectuelle

OBJET DU RAPPORT DE RECHERCHE

L'I.N.P.I. annexe à chaque brevet un "RAPPORT DE RECHERCHE" citant les éléments de l'état de la technique qui peuvent être pris en considération pour apprécier la brevetabilité de l'invention, au sens des articles L. 611-11 (nouveau) et L. 611-14 (activité inventive) du code de la propriété intellectuelle. Ce rapport porte sur les revendications du brevet qui définissent l'objet de l'invention et délimitent l'étendue de la protection.

Après délivrance, l'I.N.P.I. peut, à la requête de toute personne intéressée, formuler un "AVIS DOCUMENTAIRE" sur la base des documents cités dans ce rapport de recherche et de tout autre document que le requérant souhaite voir prendre en considération.

CONDITIONS D'ETABLISSEMENT DU PRESENT RAPPORT DE RECHERCHE

Le demandeur a présenté des observations en réponse au rapport de recherche préliminaire.

Le demandeur a maintenu les revendications.

Le demandeur a modifié les revendications.

Le demandeur a modifié la description pour en éliminer les éléments qui n'étaient plus en concordance avec les nouvelles revendications.

Les tiers ont présenté des observations après publication du rapport de recherche préliminaire.

Un rapport de recherche préliminaire complémentaire a été établi.

DOCUMENTS CITES DANS LE PRESENT RAPPORT DE RECHERCHE

La répartition des documents entre les rubriques 1, 2 et 3 tient compte, le cas échéant, des revendications déposées en dernier lieu et/ou des observations présentées.

Les documents énumérés à la rubrique 1 ci-après sont susceptibles d'être pris en considération pour apprécier la brevetabilité de l'invention.

Les documents énumérés à la rubrique 2 ci-après illustrent l'arrière-plan technologique général.

Les documents énumérés à la rubrique 3 ci-après ont été cités en cours de procédure, mais leur pertinence dépend de la validité des priorités revendiquées.

Aucun document n'a été cité en cours de procédure.

1. ELEMENTS DE L'ETAT DE LA TECHNIQUE SUSCEPTIBLES D'ETRE PRIS EN CONSIDERATION POUR APPRECIER LA BREVETABILITE DE L'INVENTION

US 6 647 528 B1 (COLLETTE WILLIAM C [US]
ET AL) 11 novembre 2003 (2003-11-11)

2. ELEMENTS DE L'ETAT DE LA TECHNIQUE ILLUSTRANT L'ARRIERE-PLAN TECHNOLOGIQUE GENERAL

WO 97/31455 A1 (ALLIED TELESYN
INTERNATIONAL C [US])
28 août 1997 (1997-08-28)

US 6 032 190 A (BREMER CLARK [US] ET AL)
29 février 2000 (2000-02-29)

3. ELEMENTS DE L'ETAT DE LA TECHNIQUE DONT LA PERTINENCE DEPEND DE LA VALIDITE DES PRIORITES

NEANT