

(12) DEMANDE INTERNATIONALE PUBLIÉE EN VERTU DU TRAITÉ DE COOPÉRATION EN MATIÈRE DE BREVETS (PCT)

(19) Organisation Mondiale de la
Propriété Intellectuelle
Bureau international



(43) Date de la publication internationale
12 décembre 2024 (12.12.2024)

WIPO | PCT

(10) Numéro de publication internationale
WO 2024/251668 A1

(51) Classification internationale des brevets :

H04L 69/04 (2022.01) H03M 7/30 (2006.01)
G06N 20/00 (2019.01) G06N 3/006 (2023.01)

(72) Inventeur : LAMPIN, Quentin ; ORANGE - Intellectual
Property and Licensing, Orange Gardens - 44 Avenue de la
République - CS 50010, 92326 Chatillon (FR).

(21) Numéro de la demande internationale :

PCT/EP2024/065211

(22) Date de dépôt international :

03 juin 2024 (03.06.2024)

(25) Langue de dépôt :

français

(26) Langue de publication :

français

(30) Données relatives à la priorité :

FR2305874 09 juin 2023 (09.06.2023) FR

(71) Déposant : ORANGE [FR/FR] ; 111, quai du Président
Roosevelt, 92130 Issy-les-Moulineaux (FR).

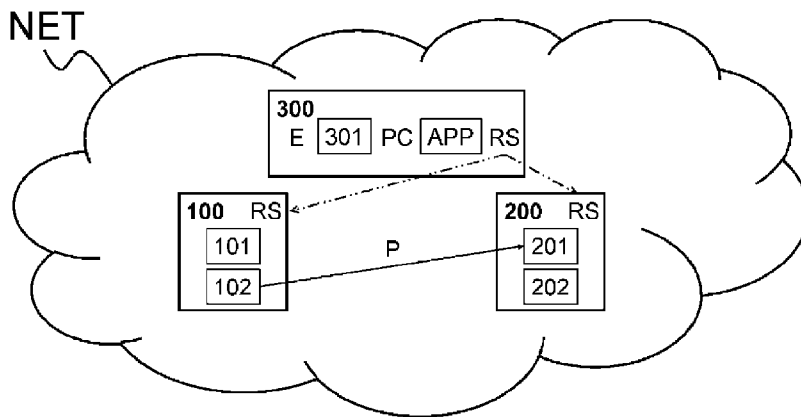
(81) États désignés (sauf indication contraire, pour tout titre de
protection nationale disponible) :

AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(54) Title: METHOD FOR OBTAINING A SET OF COMPRESSION RULES FOR A DATA PACKET TRANSMITTED IN A NETWORK

(54) Titre : PROCÉDÉ D'OBTENTION D'UN ENSEMBLE DE RÈGLES DE COMPRESSION D'UN PAQUET DE DONNÉES TRANSMIS DANS UN RÉSEAU

Fig 1



(57) Abstract: The invention relates to a method for obtaining a set (RS) of compression rules able to compress a data packet (P) transmitted in a network (NET), the packet (P) comprising sections that take respective values expressed in the form of a bit string, the packet (P) being able to be compressed by applying compression rules that are able to delete bits of a value taken by a section of the packet (P), the method comprising, for multiple given packet sections, obtaining a compression potential (PC) for a given section, in relation to the values taken by the given section in the packets belonging to a set (E) of data packets, referred to as the training set, followed by obtaining the set (RS) of compression rules on the basis of the obtained compression potentials (PC).

(57) Abrégé : L'invention se rapporte à un procédé d'obtention d'un ensemble (RS) de règles de compression aptes à compresser un paquet de données (P) transmis dans un réseau (NET), le paquet (P) comprenant des sections prenant des valeurs respectives exprimées sous la forme de suite de bits, le paquet (P) étant apte à être compressé par application de règles de compression aptes à supprimer des bits d'une valeur prise par une section du paquet (P), le procédé comprenant, pour plusieurs sections données de paquets, une obtention d'un potentiel de compression (PC) pour une section donnée, relativement aux valeurs prises par la section donnée dans les paquets appartenant à un ensemble (E) de paquets de données, dit ensemble d'apprentissage, suivie par l'obtention de l'ensemble (RS) de règles de compression en fonction des potentiels de compression (PC) obtenus.

(84) **États désignés** (*sauf indication contraire, pour tout titre de protection régionale disponible*) : ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), eurasien (AM, AZ, BY, KG, KZ, RU, TJ, TM), européen (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Publiée:

— avec rapport de recherche internationale (Art. 21(3))

Description

Titre : Procédé d'obtention d'un ensemble de règles de compression d'un paquet de données transmis dans un réseau

5 **Domaine technique**

Le domaine technique est celui de la compression et de la décompression de données.

Plus précisément, l'invention se rapporte à un procédé d'obtention d'un ensemble de règles de compression aptes à compresser un paquet de données transmis dans un réseau.

10 On verra dans la suite que l'ensemble de règles de compression obtenu selon le procédé de l'invention sera utilisé lors d'une phase de compression d'un paquet de données transmis dans un réseau.

15 Les télécommunications numériques modernes fonctionnent sur le principe de l'échange de paquets de données entre émetteurs et récepteurs suivant des protocoles tels que le protocole IP (acronyme de l'anglais *Internet Protocol*). Un ensemble de paquets de données relatifs à une communication est envoyé par l'émetteur vers un récepteur. Les paquets contiennent l'adresse du récepteur pour pouvoir lui être adressé grâce à différents algorithmes de routage, et le contenu de la communication est reconstruit par le récepteur en regroupant les données contenues dans les différents paquets. Pour réaliser une communication complète, de nombreux protocoles doivent être utilisés qui remplissent chacun un rôle depuis l'envoi physique de bits, éléments atomiques constituant les données numériques, jusqu'à l'interprétation de données pour les applications informatiques. Les bits sont classiquement regroupés en octets, ensembles de huit bits. Les
20 différents protocoles sont classiquement vus comme étant organisés en couches, un protocole d'une couche s'appuyant sur un protocole d'une couche inférieure. Le modèle OSI (de l'anglais *Open Systems Interconnection*) et le modèle Internet décrivent l'articulation des divers protocoles entre différentes couches.

25 Par exemple, dans une version simplifiée, le protocole HTTP (de l'anglais *Hypertext Transfer Protocol*) est un protocole de la couche Application, qui permet de transférer des données vers ou depuis un site web. Les données à transférer sont organisées en paquets de données. Le protocole HTTP utilise le protocole TCP (de l'anglais *Transmission Control Protocol*) qui appartient à la couche Transport, par exemple pour demander le transfert de paquets de données représentant le contenu d'un site web vers un utilisateur. Le protocole TCP, par exemple, fonctionne en trois phases : l'établissement d'une connexion, le transfert de données, et la fin de la connexion. Les protocoles de la couche Transport utiliseront un protocole de la couche Réseau, en général
30 le protocole IP (*Internet Protocol*) dans une de ses versions IPv4 ou Ipv6. Ce protocole ne s'intéresse qu'à la transmission de données organisées en paquets d'une adresse vers une autre, sans s'occuper de l'établissement de la connexion ou de l'ordre d'arrivée des paquets de données formant une transmission, ces considérations étant gérées par le protocole de la couche Transport supérieure. Le protocole IP utilisera un protocole de la couche Liaison, par exemple le protocole Ethernet qui va organiser l'envoi des données
35 élémentaires constituant le paquet, ces données élémentaires étant des bits, transmis en utilisant la couche

Physique, que ce soit en utilisant une émission par fibre optique, ou par une communication sans fil suivant les cas.

D'autres protocoles de la couche Application vont utiliser d'autres protocoles des couches inférieures pour réaliser leurs fonctions. Par exemple, le protocole CoAP (de l'anglais *Constrained Application Protocol*) utilisé dans le domaine de l'Internet des objets s'appuie en général sur le protocole UDP (de l'anglais *User Datagram Protocol*) de la couche Transport.

Un point commun à l'ensemble des protocoles sur l'ensemble des couches est l'organisation des données échangées en paquets de données. Un paquet de données est organisé en deux parties : une première partie est l'en-tête (traduction de l'anglais *header*) du paquet qui regroupe les informations nécessaires au protocole qui a la charge de traiter le paquet ; une seconde partie est la charge (traduction de l'anglais *payload*) qui regroupe les données transportées dans le paquet lui-même. Au niveau de la couche Liaison, on parle de trame plutôt que de paquet, et, en plus de l'en-tête, il existe un pied de trame indiquant la fin de la trame.

Les en-têtes de paquets comprennent des champs. Un champ est un ensemble continu de bits d'une longueur fixe ou variable. Quand un champ est de longueur variable, des mécanismes permettent de savoir quelle est la longueur du champ effectivement transmis. Quand la valeur présente dans un champ de longueur fixe est de longueur variable, des bits de bourrage (traduction de l'anglais *padding*) vont compléter la valeur jusqu'à la longueur fixe du champ. Un champ dispose d'un identifiant qui indique la nature de la valeur qui est présente dans le champ. Les valeurs présentes dans le champ sont utilisées par le protocole concerné pour traiter le paquet de façon appropriée. Par exemple, dans l'en-tête d'un paquet IP, un champ dont l'identifiant est *Version* contiendra le numéro de version du protocole utilisé. La taille de ce champ est de quatre bits. D'autres champs concernent par exemple les adresses IP source et destination du paquet. Un en-tête est donc naturellement divisé en plusieurs sections, à savoir les champs de l'en-tête. La charge du paquet peut également être divisée en plusieurs sections, par exemple en définissant des sections d'une longueur constante exprimée en nombre de bits.

Un outil d'analyse de paquets tel que Wireshark ou tcpdump permet de capturer et analyser des paquets de données. L'analyse permet de séparer un paquet de données échangé sur un réseau entre son en-tête et la charge utile, et de lire les informations comprises dans l'en-tête et les données comprises dans la charge utile et d'identifier les diverses sections d'un paquet, à la fois dans son en-tête et dans sa charge. Nous rappelons que, à la suite de l'encapsulation des paquets de données, l'en-tête typique d'un paquet sera constitué de plusieurs en-têtes concaténés, un pour chaque protocole encapsulé. Par exemple, une trame Ethernet encapsulant un paquet IP commencera par l'en-tête IP, suivi de l'en-tête UDP. Les champs du paquet IP seront donc suivis dans la trame par les champs du paquet UDP.

L'ensemble de ces éléments sont bien connus dans l'état de la technique et ne seront pas décrits plus avant.

Etat de la technique

Il existe un besoin clair de compression des données échangées dans les télécommunications numériques pour économiser au mieux les ressources matérielles et en énergie consommées par les réseaux de télécommunications. Ce besoin explique la création de protocoles dédiés à la compression des paquets de données échangées.

Les paquets de données échangées sont constitués, comme on l'a vu, d'une part d'un en-tête contenant les informations nécessaires à la transmission du paquet et d'autre part d'une charge utile comprenant les données transmises par le paquet en tant que telles. La compression peut donc porter sur la charge utile, ou bien sur les en-têtes de paquets de données. Comme les valeurs de certains champs des en-têtes de paquets de données vont prendre des valeurs constantes ou variant peu, il est possible de procéder à des compressions très efficaces des paquets de données en cherchant uniquement à comprimer les valeurs des champs des en-têtes des paquets de données.

Un premier exemple de protocole comprimant les valeurs des champs d'en-têtes de paquets de données est le protocole ROHC (acronyme de l'anglais *Robust Header Compression*) qui est défini par l'organisme de standardisation IETF (pour *Internet Engineering Task Force*) dans le document RFC (*Request For Comments*, documents de spécification à la base d'Internet) numéro 3095. L'idée de base de ce protocole est que certaines valeurs des champs des en-têtes des paquets formant une même communication ne vont pas changer d'une communication sur l'autre. Par exemple, les valeurs des champs d'adresse source et destination vont rester les mêmes pendant une communication. Après un envoi d'un premier paquet contenant les valeurs de ces champs, ces valeurs seront effacées dans les en-têtes des paquets suivants afin de comprimer l'en-tête du paquet, et donc le paquet lui-même, et seront insérées à la réception puisqu'elles sont connues après l'envoi du premier paquet. L'effacement est effectué par un module de compression et l'insertion de la valeur par un module de décompression. D'autres types de compression peuvent être appliqués.

On obtient ainsi une compression très efficace des paquets de données même si la compression se limite aux en-têtes. Le protocole ROHC est utilisé par exemple dans les réseaux de télécommunications mobiles. Mais il nécessite que soit construit pour chaque communication un contexte indiquant quelles valeurs peuvent être comprimées, et de quelle manière. Les émetteurs et les récepteurs doivent donc mettre à jour régulièrement les contextes utilisés pour la compression des en-têtes. Cette opération de mise à jour est très coûteuse et d'autres mécanismes vont être préférés dans certains contextes d'utilisation.

Le contexte des réseaux de communication dédiés aux objets connectés impose des contraintes supplémentaires. On parle ici de réseaux étendus à basse consommation, également dénommés LPWANs, acronyme de l'anglais *Low-Power Wide Area Networks*. Dans un tel réseau, on retrouve des objets connectés qui doivent économiser au maximum leur consommation électrique, comme des compteurs déployés à des endroits n'ayant pas d'alimentation électrique et devant utiliser une batterie pendant plusieurs années. Ces objets communiquent sans fil avec une passerelle qui couvre une zone étendue. La mise à jour du contexte à

chaque communication impliquerait une consommation de bande passante élevée, si bien que le protocole ROHC de compression d'en-têtes n'est pas adapté dans ce cadre.

De plus, les données échangées par ces objets sont souvent de taille très réduite, et peuvent se limiter à signaler que l'objet est toujours actif, ou bien l'objet peut vouloir remonter une seule valeur, telle qu'une température pour un thermomètre, ou un niveau d'eau pour un compteur d'eau. La taille des en-têtes des paquets de données proportionnellement à la taille de la charge utile va donc être proportionnellement plus grande dans le contexte des LPWANs que dans d'autres contextes. Il existe donc un besoin de compression efficace des paquets de données qui, dans le contexte des LPWANs, va se porter sur la compression des valeurs des champs des en-têtes.

Le protocole SCHC (acronyme de l'anglais *Static Context Header Compression*), défini par l'IETF dans la RFC 8724, a été conçu pour être utilisé dans le contexte des LPWANs. Il se fonde sur un contexte statique. Pour un contexte donné, par exemple un type d'objet connecté déployé dans un réseau donné, des règles de compression vont être définies. Ces règles regroupent des opérations qui peuvent être appliquées aux valeurs des champs des en-têtes de paquets. Ces opérations peuvent être de trois types, toutes fondées sur l'idée que des valeurs de certains champs, ou une partie des valeurs, peuvent être effacées avant transmission puis reconstruites à la réception :

- Un premier type d'opération de compression est l'effacement : la valeur du champ est lue et sera effacée par le module de compression si la valeur est bien celle attendue. Le module de compression est par exemple situé dans l'objet connecté. La valeur effacée sera insérée à la réception par le module de décompression qui, lui, est situé dans la passerelle et qui dispose des mêmes règles que le module de compression. Ce type d'opération est adapté aux champs qui prennent une valeur constante, comme le champ *Version* des paquets IP. Si le champ prend une autre valeur que celle attendue, l'opération ne s'applique pas, la règle de compression est ignorée, et une autre règle sera éventuellement utilisée ou bien la valeur du champ pourra être transmise inchangée.
- Un deuxième type d'opération de compression utilise un dictionnaire : un dictionnaire est construit pour certaines valeurs du champ et si la valeur du champ se trouve dans le dictionnaire, le module de compression va remplacer la valeur du champ par son index dans le dictionnaire. Le module de décompression utilise le dictionnaire pour remplacer l'index par la valeur effacée. Là encore, une valeur de champ en-dehors du dictionnaire pourra être transmise inchangée si aucune règle ne s'applique.
- Un troisième type d'opération consiste en l'effacement du préfixe : la valeur du champ est une suite de bits, et un préfixe plus ou moins long peut être défini dans une opération de compression. Si la valeur a bien ce préfixe, le préfixe sera effacé à la compression et inséré à la décompression, ou bien la valeur sera transmise inchangée si le préfixe n'est pas reconnu par l'opération.

Pour des raisons de complétion, un quatrième type d'opération est défini qui consiste à ignorer la valeur du champ et à la transmettre telle quelle.

D'autres combinaisons sont possibles, ainsi que d'autres actions de compression/ décompression, par exemple pour traiter les champs de longueur variable.

Un ensemble de règles de compression va alors être défini pour un contexte de déploiement donné. Par exemple, des règles de compression (et leurs règles symétriques de décompression) peuvent être définies pour une passerelle gérant un réseau LPWAN et pour l'ensemble des équipements connectés à ce réseau et à Internet grâce à la passerelle. L'ensemble de règles pourra alors traiter en compression et décompression l'ensemble des paquets de données qui s'échangent dans le réseau LPWAN, la compression se limitant aux valeurs des champs des en-têtes. Le protocole SCHC permet de déterminer quelle règle de compression s'applique lorsqu'un paquet de données circule dans le réseau.

Pour cela, chaque règle de compression présente dans l'ensemble est définie pour un format donné d'en-tête de paquet et va consister en un ensemble d'opérations à appliquer sur les valeurs des champs de l'en-tête d'un paquet qui respecte le format. Plusieurs règles peuvent couvrir un même format d'en-tête de paquet de données. Par exemple, des règles de compression vont être définies pour le format des paquets IP encapsulant un paquet UDP, et ces règles comprendront des opérations pour tous les champs d'un en-tête de paquet IP suivi par un en-tête de paquet UDP.

Une règle de compression fait partie d'un ensemble de règles et dispose d'un identifiant de règle. L'ensemble de règles va chercher à couvrir l'ensemble des formats possibles d'en-têtes de paquets de données qui peuvent être échangés dans un contexte donné.

Le protocole SCHC de compression/décompression est appliqué par un module de compression qui va traiter un paquet avant de l'adresser à un module de décompression. Les modules de compression et de décompression disposent du même ensemble de règles, défini pour un contexte donné. Le protocole va consister, lorsqu'un paquet de données doit être traité par le module de compression, à parcourir l'ensemble de règles dont dispose le module de compression afin de retenir une ou plusieurs règles. Si une règle ne correspond pas au format des champs de l'en-tête du paquet traité, la règle n'est pas retenue. Si le format des champs correspond, il faut examiner si les valeurs des champs correspondent à tous les opérateurs de correspondance prévus dans la règle. Si ce n'est pas le cas, la règle là encore n'est pas retenue. Si aucune règle de l'ensemble ne correspond, le paquet est expédié sans traitement. Dans tous les cas, la charge utile du paquet n'est pas traitée par le protocole SCHC. Dans le contexte des LPWANs, la compression du paquet limitée à la compression des valeurs des champs des en-têtes sera quand même efficace, car la taille des en-têtes est proportionnellement importante.

On voit ici l'intérêt de l'opération qui consiste à ignorer la valeur du champ et à la transmettre telle quelle. Cette opération permet de s'assurer qu'une règle pourra effectivement s'appliquer à tous les champs d'un en-tête donné : pour les champs faciles à comprimer, une opération de compression efficace sera définie, et pour les autres champs l'opération qui ne fait rien servira de complétion.

Finalement, plusieurs règles peuvent être retenues pour lesquelles le format des champs d'une part et les valeurs des champs d'autre part correspondent. Une règle est alors choisie, et une seule, par exemple la

première rencontre dans l'ensemble de règles, ou bien choisie aléatoirement, au choix de l'implémentation du protocole SCHC. Les opérations prévues dans la règle sont appliquées par le module de compression aux valeurs des champs de l'en-tête du paquet. Puis le paquet est transmis avec les valeurs de son en-tête comprimées, ainsi qu'avec l'identifiant de la règle qui a été appliquée. Le module de décompression reçoit le paquet dont les valeurs de l'en-tête ont été comprimées ; utilise l'identifiant de la règle pour trouver les opérations qui ont été appliquées ; et applique les opérations symétriques de décompression. Par exemple, si, pour une valeur d'un champ, l'opération de compression a utilisé un dictionnaire, le module de décompression utilisera la valeur de l'index transmise pour le remplacer par la valeur prévue dans le dictionnaire.

Comme l'identifiant de la règle utilisée pour la compression doit être transmis, ainsi que certaines valeurs (index des dictionnaires), il va exister une limite au-delà de laquelle ajouter de nouvelles règles de compression dans un contexte donné ne sera plus intéressant car le gain éventuel qu'apporte la règle supplémentaire est dépassé par le coût de transmission induit par l'augmentation de la taille de l'identifiant de règle au fur et à mesure que de nouvelles règles sont ajoutées. Par exemple, au-delà de huit règles, l'identifiant de règle transmis sera codé sur trois bits au lieu de deux. Si rajouter une neuvième règle ne fait gagner en moyenne qu'un seul bit, l'ajout de cette neuvième règle n'est pas intéressant. De même si une règle ajoutée implique l'utilisation de bits supplémentaires pour encoder les index de dictionnaires.

Il est facile de voir que les opérations de compression et de décompression dans le protocole SCHC peuvent être sans perte, c'est-à-dire que la décompression de l'en-tête du paquet de données selon les opérations prévues dans le protocole SCHC reconstruit exactement l'en-tête tel qu'il était avant compression. C'est le cas général même s'il peut exister des cadres d'utilisation où ce n'est pas le cas.

Les ensembles de règles de compression sont donc définies, dans le protocole SCHC, pour un contexte donné, statique, et correspondent aux formats des paquets attendus dans les échanges entre émetteurs et récepteurs. Ils sont définis par un expert qui sait que, dans un contexte de déploiement donné, tel ou tel protocole va être utilisé, selon une encapsulation donnée, et qui sait donc quels formats des en-têtes des paquets sont attendus. L'expert sait également que tel ou tel champ des paquets respectant tel ou tel format prendra une valeur constante, ou bien une valeur parmi un ensemble de taille limitée, ou bien une valeur qui aura souvent le même préfixe. Par exemple, l'expert sait que le champ *IP-Version* qui apparaîtra dans les en-têtes des paquets respectant plusieurs formats (IP seul, ou IP encapsulant des paquets UDP, ou autre...) sera constante. Ou bien l'expert sait que les valeurs d'adresses IP présentes dans un champ donné appartiennent à une plage limitée et auront donc un préfixe commun. Cette connaissance permet à l'expert de choisir l'opération la plus appropriée pour un champ donné dans un format donné et de l'intégrer dans la règle. L'expert construit donc un ensemble de règles qui sera déployé, par exemple attaché à des modules de compression et de décompression embarqués dans un ensemble d'objets connectés et une passerelle correspondante, pour mettre en œuvre le protocole SCHC.

Le protocole SCHC a été initialement conçu dans le contexte des réseaux LPWANs mais il pourrait être utilisé dans d'autres contextes, comme les réseaux domestiques, ou pour la compression d'en-têtes de trames

Ethernet. Les opérations de compression du protocole SCHC pourraient aussi être appliquées à des sections de la charge d'un paquet de données et pas seulement à l'en-tête.

Dans la situation actuelle, le déploiement du protocole SCHC est ralenti par le besoin de créer de nouvelles règles pour chaque contexte d'utilisation. Cette situation fait que des potentiels de compression facilement accessibles ne sont pas utilisés. Ceci a des impacts négatifs sur les consommations énergétiques et matérielles des réseaux LPWANs ou celles d'autres réseaux dans lesquels la compression des en-têtes selon le protocole SCHC pourrait être utilisée.

Une autre limite de la situation actuelle est que les règles de compression définies pour un contexte de déploiement donné ne vont plus aussi bien fonctionner quand ce contexte évolue. Par exemple, si un objet connecté redémarre, son adresse réseau peut changer ou bien les numéros de port qu'il utilise, et des règles de compression qui étaient efficaces peuvent se retrouver inefficaces. C'est un point qui introduit de l'inefficacité dans les compressions qui peuvent être obtenues.

De plus, il est souhaitable d'utiliser les opérations de compression définies par le protocole SCHC, qui apportent une compression très efficace, dans un maximum de contextes. Par exemple, ces opérations de compression devraient pouvoir s'appliquer à des sections de la charge du paquet, et pas seulement à des champs d'en-tête. Or, il n'existe pas à l'heure actuelle de solution technique permettant d'appliquer les principes de compression d'en-têtes définis dans le protocole SCHC à un contexte dynamique, même si celui-ci est faiblement dynamique. Par exemple, si, dans un contexte de déploiement, la compression SCHC s'applique à une valeur d'adresse IP, et que cette valeur vient à changer même si c'est à une fréquence très faible, la règle établie dans un contexte statique devient inopérante dès le premier changement d'adresse IP. Ou bien, si une section donnée d'une charge varie, même peu, les règles du protocole SCHC ne vont pas s'appliquer.

L'invention vient améliorer la situation.

Exposé de l'invention

Selon un premier aspect fonctionnel, l'invention a trait à un procédé d'obtention d'un ensemble de règles de compression d'un paquet de données transmis dans un réseau, le paquet comprenant des sections prenant des valeurs respectives exprimées sous la forme de suite de bits, le paquet étant apte à être compressé par application de règles de compression aptes à supprimer des bits d'une valeur prise par une section du paquet, le procédé comprenant, pour plusieurs sections données de paquets, une obtention d'un potentiel de compression pour une section donnée, relativement aux valeurs prises par la section donnée dans les paquets appartenant à un ensemble de paquets de données, ensemble dit d'apprentissage, suivie par l'obtention de l'ensemble de règles de compression en fonction des potentiels de compression obtenus.

Grâce à l'invention, un ensemble de règles de compression pertinentes en terme de compression est créé automatiquement. Les règles de compression comprennent des opérations de compression qui s'appliquent à

des valeurs prises par des sections de paquets de données. Les règles de compression de l'ensemble obtenu peuvent être appliquées aux paquets de données transmis dans le réseau afin de réaliser une compression effective et efficace des paquets de données transmis. L'avantage de l'invention est que les règles de compression peuvent changer automatiquement quand le contexte de transmission change et rester efficaces. Cela permet de faciliter le déploiement de protocoles de compression et de les utiliser dans de nombreux contextes, y compris des contextes dynamiques.

Dans l'invention, des grandeurs dites potentiels de compression sont obtenues à partir d'un ensemble de paquets, ensemble dit d'apprentissage. Cet ensemble dit d'apprentissage est représentatif du trafic attendu, qui devra être comprimé par l'ensemble de règles de compression obtenu. On verra dans la suite que l'obtention de potentiels de compression permet d'identifier les sections les plus prometteuses à comprimer dans cet ensemble représentatif et ainsi de guider la création de règles de compression. L'obtention de potentiels de compression repose sur les fréquences d'apparition de valeurs différentes dans l'ensemble dit d'apprentissage, ce qui est une grandeur facile à obtenir même pour de très grands ensembles dit d'apprentissage.

Selon un mode de réalisation, le procédé comprend de surcroît l'application d'au moins une opération de compression à une valeur de section dudit paquet de données, ladite opération étant comprise dans une règle de compression appartenant à l'ensemble de règles de compression obtenu, suivie par la transmission dudit paquet de données auquel ladite au moins une opération de compression a été appliquée et d'une donnée représentative de la règle de compression à laquelle appartient ladite opération.

Grâce à ce mode de réalisation, les règles de compression obtenues sont appliquées à des paquets de données et permettent de réaliser des compressions effectives. Les paquets de données sont comprimés par l'application d'opérations de compression qui effacent des portions de valeur, voire des valeurs complètes, prises par des sections du paquet de données. Pour reconstruire les portions de valeur effacées à la réception du paquet comprimé, celui-ci doit être transmis avec une donnée représentative de la règle de compression qui lui a été appliquée.

Selon un mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, la section du paquet de données à la valeur de laquelle est appliquée une opération de compression est un champ d'un en-tête du paquet de données.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, la section du paquet de données à la valeur de laquelle est appliquée une opération de compression est une délimitation définie au préalable de la charge du paquet de données.

Ces deux modes de réalisation permettent d'appliquer l'invention à l'ensemble des valeurs prises par les sections du paquet de données, qu'il s'agisse de champs d'en-têtes ou bien de parties délimitées et définies au préalable de la charge du paquet. Dans ce dernier cas, le fait que la délimitation soit définie au préalable permet de faciliter l'application du procédé. Une opération donnée, retrouvée grâce à la donnée représentative transmise, va par exemple s'appliquer aux 256 premiers bits de la charge, et le paquet pourra

être rétabli à l'original en utilisant la connaissance de cette définition au préalable de la délimitation. Une autre délimitation définie au préalable peut par exemple consister à s'appliquer à la totalité de la charge du paquet, ou bien à la première ou seconde moitié, ou à toute autre proportion définie au préalable.

5 Les champs d'un en-tête sont eux-mêmes des sections définies au préalable puisque la structure de l'en-tête, et donc sa délimitation en plusieurs champs, est définie par le protocole, ou l'empilement de protocoles, auquel appartient le paquet de données transmis.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, le paquet de données auquel ladite au moins une opération de compression a été appliquée est transmis conjointement avec la donnée représentative de la règle de compression à laquelle
10 appartient ladite opération de compression.

Grâce à ce mode de réalisation, le paquet de données est transmis conjointement avec la donnée représentative de la règle de compression qui lui a été appliquée. Cette transmission conjointe permettra de faciliter l'opération de décompression ultérieure qui sera appliquée au paquet de données, en s'assurant que, lorsqu'un paquet de données est transmis, il est transmis avec l'information qui permet de connaître
15 l'opération de compression qui lui a été appliquée.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, le paquet de données transmis est formé en concaténant une donnée représentative d'une règle de compression, ladite donnée étant un identifiant de la règle de compression consistant en une suite de bits, avec le paquet de données auquel a été appliqué au moins une opération de compression
20 appartenant à ladite règle de compression qui appartient à un ensemble de règles de compression obtenu par apprentissage.

Grâce à ce mode de réalisation, un format précis du paquet de données compressé est défini qui va permettre de faciliter la transmission de celui-ci. Le paquet de données transmis est un nouveau paquet de données formé en concaténant un identifiant de la règle de compression, à savoir une suite de bits, avec le
25 paquet de données original auquel a été appliqué la règle de compression. Le paquet de données compressé est donc transmis conjointement avec un identifiant de la règle de compression qui permettra de retrouver celle-ci parmi l'ensemble de règles de compression obtenu par apprentissage.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, l'obtention de l'ensemble de règles de compression comprend la mise en œuvre d'un
30 algorithme d'apprentissage par renforcement.

L'apprentissage par renforcement, au contraire de l'apprentissage supervisé, ne nécessite aucune intervention d'expert. L'effet technique obtenu est donc que les règles de compression peuvent être obtenues automatiquement afin de réaliser la compression des paquets de données transmis dans le réseau. Un premier avantage est que les règles de compression peuvent être obtenues à bas coût et rapidement puisque

cette obtention ne fait pas intervenir d'expert. Un autre avantage est que les règles de compression peuvent changer automatiquement quand le contexte de transmission change et rester efficaces.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, l'algorithme d'apprentissage par renforcement mis en œuvre dans l'obtention de l'ensemble' de règles de compression utilise comme ensemble d'apprentissage l'ensemble dit d'apprentissage, formé de paquets de données collectés au préalable ; met en œuvre un environnement qui comprend l'ensemble dit d'apprentissage et l'ensemble de règles de compression ; et le changement d'état dans l'algorithme d'apprentissage par renforcement comprend l'ajout d'une nouvelle règle à l'ensemble de règles de compression.

Comme l'invention s'appuie dans ce mode de réalisation sur l'utilisation d'un algorithme d'apprentissage, l'invention nécessite d'utiliser un ensemble d'apprentissage qui doit être défini au préalable. Cet ensemble d'apprentissage est un ensemble de paquets de données pour lequel l'algorithme d'apprentissage va chercher à produire un ensemble de règles de compression qui le comprime au mieux. Cet ensemble d'apprentissage est l'ensemble dit d'apprentissage qui est utilisé pour obtenir les potentiels de compression d'une section donnée d'un paquet selon les valeurs prises par la section donnée dans les paquets de l'ensemble dit d'apprentissage. Il est clair que l'apprentissage sera d'autant plus pertinent que l'ensemble d'apprentissage sera représentatif du trafic attendu dont on doit compresser les paquets de données. Si l'ensemble d'apprentissage est trop petit, en particulier, l'apprentissage ne pourra pas obtenir un ensemble de règles de compression efficaces dans tous les cas, car des exemples de trafic qui auraient pu être comprimés ne sont pas présents dans l'ensemble d'apprentissage. L'apprentissage par renforcement consiste à faire évoluer l'ensemble de règles de compression qui est appris, dans un environnement comprenant l'ensemble d'apprentissage que l'ensemble de règles de compression doit compresser. Cette évolution se fait par l'ajout d'une nouvelle règle à l'ensemble de règles de compression. L'ensemble d'apprentissage, quant à lui, n'évolue pas durant la phase d'apprentissage.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, l'algorithme d'apprentissage par renforcement comprend une fonction de récompense qui est calculée selon la compression des paquets de l'ensemble d'apprentissage par les règles de compression appartenant à l'ensemble de règles de compression.

Il est avantageux que plusieurs paramètres de l'algorithme d'apprentissage par renforcement soient définis afin de pouvoir l'exécuter. Un apprentissage par renforcement consiste à faire évoluer un acteur, ici l'ensemble de règles de compression que l'on apprend, dans un environnement, ici la réunion de l'ensemble de règles et de l'ensemble d'apprentissage, et de guider les évolutions de l'acteur grâce à une fonction de récompense. La fonction de récompense est la compression réalisée par l'ensemble de règles de compression sur les paquets de l'ensemble d'apprentissage. Meilleure est la compression réalisée par l'ensemble de règles, meilleur est cet ensemble et donc l'apprentissage par renforcement aura tendance à produire un ensemble de règles qui produit une meilleure compression. Comme déjà vu, le changement d'état est l'ajout d'une règle à l'ensemble de règles ; la fonction de récompense viendra sanctionner si cet ajout de

règle est pertinent ou non et permet bien de faire évoluer l'ensemble de règles de compression vers une meilleure compression.

L'avantage de ce mode est donc de préciser comment exécuter un algorithme d'apprentissage par renforcement pour apprendre un ensemble de règles de compression et comment choisir des paramètres pertinents de l'apprentissage par renforcement pour obtenir une bonne compression des paquets de l'ensemble d'apprentissage par l'ensemble de règles résultat de l'apprentissage.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, l'apprentissage d'un ensemble de règles de compression est réalisé à partir d'un ensemble initial consistant en une règle unique ne réalisant aucune compression.

Dans ce mode de réalisation, le point de départ est un ensemble de règles formé d'une règle unique s'appliquant à tous les paquets de l'ensemble d'apprentissage collecté. Cet ensemble permet de s'assurer que dès le départ, l'ensemble de règles va bien s'appliquer à tout l'ensemble d'apprentissage même si la règle initiale ne réalise pas de compression sur les valeurs de sections des paquets de l'ensemble d'apprentissage. Dans la plupart des modes de réalisation de notre invention, l'ensemble de règles de compression évolue de manière à s'assurer que, pour tous les paquets de l'ensemble d'apprentissage, il existe dans l'ensemble de règles de compression une règle qui puisse s'appliquer, ce qui est bien le cas avec cette règle unique ne réalisant aucune compression.

L'avantage de ce mode de réalisation est de ne nécessiter l'intervention d'aucun expert pour définir l'ensemble initial de règles de compression qui est soumis à l'algorithme d'apprentissage.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, l'apprentissage d'un ensemble de règles de compression est réalisé à partir d'un ensemble initial comprenant une règle de compression fournie par un expert.

L'avantage de ce mode de réalisation est de permettre à un expert de s'assurer qu'une règle qui lui semble très utile est bien présente dans l'ensemble de règles de compression. En la plaçant dès le départ de l'apprentissage, on obtient ce résultat. Le reste de l'algorithme d'apprentissage pourra se faire par renforcement, donc sans intervention ultérieure de l'expert. On obtient donc ici un mode mixte où l'intervention de l'expert est limitée.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, le procédé de gestion comprend les étapes suivantes exécutées lors de l'obtention de l'ensemble de règles de compression :

- une étape de sélection d'une règle appartenant à l'ensemble de règles de compression en utilisant au moins un potentiel de compression parmi ceux obtenus ;
- une étape de création d'une nouvelle règle à partir de la règle sélectionnée en utilisant au moins un potentiel de compression parmi ceux obtenus ;
- l'ajout de la nouvelle règle à l'ensemble de règles de compression ;

et, de plus, lesdites étapes sont itérées, jusqu'à ce qu'un critère d'arrêt soit satisfait, en utilisant à l'itération suivante l'ensemble de règles modifié par l'ajout de la nouvelle règle.

Grâce à ce mode de réalisation, l'obtention d'un ensemble de règles de compression est guidé de façon efficace. En effet, l'obtention peut fonctionner en réalisant aléatoirement des changements d'état qui seront

5 retenus ou non selon une fonction de récompense. Cette manière de procéder pourrait d'une part prendre beaucoup de temps et d'autre part conduire à des ensembles de règles de compression comprenant un nombre trop élevé de règles si les règles les plus efficaces ne sont pas créées en premier. Or, on a vu que si un trop grand nombre de règles de compression est créé, cela devient un obstacle à une compression efficace car l'index de la règle utilisée lors d'une compression doit être transmise avec le paquet comprimé.

10 Un trop grand nombre de règles implique un index codé sur un grand nombre de bits qui rend inefficace la compression.

Il faut donc guider l'obtention des règles de compression. Pour cela, les potentiels de compression obtenus vont d'abord indiquer quelle règle est sélectionnée pour créer une nouvelle règle puis comment créer une nouvelle règle à partir de la règle sélectionnée.

15 L'obtention fonctionne par itération jusqu'à ce qu'un critère d'arrêt soit satisfait, et l'itération consiste à appliquer le même algorithme à un environnement formé du nouvel ensemble de règles de compression (avec une nouvelle règle ajoutée) et de l'ensemble dit d'apprentissage. Pour s'assurer que l'obtention prend bien en compte l'ajout de la nouvelle règle, il peut être utile de calculer à l'itération suivante des potentiels de compression qui prennent bien en compte l'ajout de la nouvelle règle. Cela peut se faire en isolant dans

20 l'ensemble dit d'apprentissage les sous-ensembles de paquets pour lesquels les différentes règles s'appliquent. Dans ce mode, les valeurs comprimées par la nouvelle règle ne seront pas prises en compte dans les calculs de potentiels de compression. Les potentiels de compression ne seront calculés que pour les valeurs qui ne sont pas comprimées dans l'état courant de l'ensemble de règles de compression qui est en cours d'obtention.

25 Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, si la règle ajoutée s'applique à tous les paquets de données de l'ensemble dit d'apprentissage auxquels la règle sélectionnée s'applique, alors la règle sélectionnée est retirée de l'ensemble de règles de compression.

Il est avantageux que l'ajout de la nouvelle règle assure que l'ensemble de règles qui est en cours d'obtention

30 va bien s'appliquer à l'ensemble des paquets de l'ensemble dit d'apprentissage. C'est pour cela que, dans ce mode de réalisation, si la règle ajoutée ne couvre pas l'ensemble des paquets que couvrirait la règle sélectionnée à partir de laquelle a été créée la règle ajoutée, alors la règle sélectionnée est conservée. Comme l'algorithme d'obtention peut partir, dans certains modes, d'un ensemble comprenant une règle initiale s'appliquant à tous les paquets, on est sûr alors de bien disposer à la fin d'un ensemble de règles

35 s'appliquant à tous les paquets de l'ensemble dit d'apprentissage. Cependant, la suppression de la règle sélectionnée si la règle ajoutée couvre les mêmes paquets permet de s'assurer que des règles inutiles ne

sont pas conservées et donc de minimiser la taille de l'ensemble de règles de compression résultat de l'algorithme d'apprentissage.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, le critère d'arrêt est satisfait quand la compression obtenue sur l'ensemble dit d'apprentissage par l'ensemble modifié de règles de compression est inférieure ou égale à la compression obtenue par l'ensemble de règles précédent.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, le critère d'arrêt est satisfait quand l'ensemble de règles atteint une taille prédéterminée exprimée en nombre de règles ou en nombre d'opérations de compression incluses dans les règles.

Ces deux modes ont l'intérêt de proposer un critère d'arrêt facile à calculer. Dans le premier mode, la compression réalisée par l'ensemble de règles sur l'ensemble dit d'apprentissage est calculée et l'obtention s'arrête quand la compression ne progresse plus. Cela va arriver par exemple quand toutes les valeurs des sections des paquets de l'ensemble dit d'apprentissage sont comprimées. Dans ce cas, il n'est plus possible de rajouter une opération de compression. La compression réalisée par le nouvel ensemble peut être inférieure à la compression réalisée par l'ensemble précédent car il faut prendre en compte que les règles de compression doivent être identifiées, et que la taille de l'index de la règle ajoutée peut être plus grand que la compression réalisée par cette règle. Dans le deuxième mode, on utilise un critère de taille absolue de l'ensemble de règles de compression, soit en nombre de règles soit en nombre d'opérations. Ce mode permet de s'assurer que la taille de l'ensemble de règles obtenu reste toujours inférieure à une certaine taille, ce qui est avantageux dans un contexte où l'on veut économiser les ressources et où on veut s'assurer que les index des règles de compression pourront être encodés avec un nombre limité de bits (de deux à quatre par exemple). Ce mode de réalisation est particulièrement avantageux quand l'ensemble de règles de compression obtenu par le procédé est utilisé par des équipements aux capacités mémoire limitées, qui ne pourront mémoriser qu'un nombre limité de règles de compression de données émises ou reçues. Les deux modes peuvent être combinés, à savoir que l'apprentissage se poursuit tant que la compression réalisée par l'ensemble de règles progresse sauf si une taille maximale prédéfinie est atteinte.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, l'un des potentiels de compression obtenu est dit potentiel de compression de champ et est calculé pour une section donnée de longueur fixe suivant la formule $N(L - H)$ dans laquelle N est le nombre de fois que la section apparaît dans l'ensemble dit d'apprentissage ; L est la longueur de la section exprimée en bits et H est une grandeur dite estimateur d'entropie de champ qui est calculée suivant la formule $H = - \sum_{i \in [1, K]} f_i \log_2(f_i)$ dans laquelle K est le nombre de valeurs différentes $V_{i \in [1, K]}$ que prend la section dans l'ensemble dit d'apprentissage et f_i est la fréquence d'apparition de la valeur V_i .

Dans de nombreux modes de réalisations, les sections pour les valeurs desquelles l'algorithme d'obtention va chercher à produire des opérations de compression appartenant à des règles sont des champs d'en-têtes. Ceci explique le choix des termes « potentiel de compression de champ » et « estimateur d'entropie de

champ ». Cependant, le mode de réalisation peut s'appliquer aussi bien à des champs d'en-tête qu'à des sections délimitées dans la charge d'un paquet de données.

La formule de calcul dans ce mode de réalisation s'applique à une section dont la longueur L , exprimée en bits, est constante pour toutes les valeurs prises par la section dans l'ensemble dit d'apprentissage. On verra plus loin qu'un autre mode de réalisation peut s'appliquer à des sections de taille variable.

Grâce à ce mode de réalisation, le potentiel de compression de champ est calculé pour une section donnée, afin de détecter quelle est la section la plus intéressante à comprimer et guider ainsi l'algorithme d'obtention.

Pour cela, nous utilisons une grandeur qui estime l'entropie présente dans une section délimitée des charges des paquets de l'ensemble dit d'apprentissage ou dans un champ donné des en-têtes des paquets de l'ensemble dit d'apprentissage, que nous appelons section en général. Nous rappelons qu'une entropie (au sens de la théorie de l'information) se calcule à partir de probabilités. Ici, nous disposons d'un ensemble dit d'apprentissage, donc de valeurs observées. C'est pour cela que nous calculons un estimateur d'entropie, et non une entropie proprement dite, à partir des fréquences d'apparition f_i des valeurs V_i que prend la section considérée. Ces fréquences sont strictement positives et inférieures ou égales à 1 par définition. Les logarithmes des fréquences sont pris en base 2 pour s'appliquer à des valeurs exprimées en bits. Les logarithmes sont forcément négatifs ou nuls puisque les fréquences valent 1 au maximum. Les négations rendent l'estimateur d'entropie forcément positif.

L'estimateur d'entropie peut donc valoir au minimum zéro. C'est le cas quand la section donnée présente une seule valeur dans l'ensemble dit d'apprentissage. La fréquence d'apparition de cette valeur unique vaut donc 1 ; et le logarithme vaut 0.

Par ailleurs, l'estimateur d'entropie sera toujours inférieur à la longueur de la section exprimée en bits. Si une section délimitée, ou un champ, prend un grand nombre de valeurs différentes, et donc si son estimateur d'entropie augmente, alors la longueur de la section exprimée en bits qui est nécessaire pour encoder ce grand nombre de valeurs différentes va également croître et restera toujours supérieure à l'estimateur d'entropie.

La grandeur dite potentiel de compression de champ est donc toujours positive et sera maximale pour une section donnée quand son estimateur d'entropie est nul. Cela arrive quand la section prend une valeur unique dans l'ensemble dit d'apprentissage et on comprend facilement qu'une section qui prend une valeur unique va être facile à comprimer. Il suffit d'effacer cette valeur ! Le potentiel de compression de champ diminue au fur et à mesure que l'estimateur d'entropie augmente.

Si une section dont l'estimateur d'entropie est nul n'apparaît que peu de fois dans l'ensemble d'apprentissage, ou bien si la longueur de la section est faible, une autre section dont l'estimateur d'entropie est plus élevé mais qui apparaît plus souvent, ou dont la longueur est plus grande, pourra être plus intéressant à comprimer que la section à l'estimateur d'entropie nulle. C'est pour cela que le potentiel de compression du champ prend en compte le nombre de fois que la section apparaît ainsi que sa longueur.

Le potentiel de compression de champ permet donc de guider de façon efficace l'algorithme d'obtention en indiquant les sections de l'ensemble d'apprentissage dont la compression rapportera le plus selon la fonction de récompense qui est donnée par la compression réalisée, lesdites sections pouvant être des champs d'entêtes des paquets ou des sections délimitées de la charge des paquets.

- 5 Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, le potentiel de compression de champ est calculé pour une section donnée de longueur variable suivant la formule $\sum_{i=1}^K N_i \cdot (L_i - H_i)$ dans laquelle K est le nombre de valeurs différentes $V_{i \in [1, K]}$ que prend la section dans l'ensemble dit d'apprentissage ; N_i est le nombre de fois que la section prend la valeur V_i dans l'ensemble dit d'apprentissage ; L_i est la longueur de la valeur V_i exprimée en bits et H_i est une
- 10 grandeur dite estimateur d'entropie partielle d'une valeur, qui est calculée pour la valeur V_i suivant la formule $H_i = -f_i \log_2(f_i)$ dans laquelle f_i est la fréquence d'apparition de la valeur V_i dans l'ensemble dit d'apprentissage.

- Grâce à ce mode de réalisation, le potentiel de compression de champ peut être calculé pour une section de longueur variable. Plusieurs éléments vont donc être pris en compte dans ce calcul. Une valeur qui apparaît
- 15 fréquemment, mais qui est de faible longueur, pourra moins contribuer au potentiel de compression de champ qu'une valeur qui apparaît moins fréquemment mais dont la longueur est plus grande. Ce mode de réalisation permet donc de réaliser un calcul précis du potentiel de compression de champ même pour une section de longueur variable. On peut remarquer que la formule utilisée dans ce mode donne les mêmes résultats que la formule donnée précédemment dans le cas où la section a une longueur L constante pour toutes les valeurs
- 20 prises par la section.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, le potentiel de compression de champ est calculé suivant la formule $N \frac{L}{H+a}$ dans laquelle a est une constante strictement positive.

- Ce mode de réalisation donne une variante pour obtenir un potentiel de compression de champ. Dans cette
- 25 formule, le potentiel est bien maximal quand l'estimateur d'entropie est nul. La constante positive permet d'assurer la définition du potentiel de compression de champ dans ce cas. Cette variante est adaptée au cas où la section est de longueur fixe.

- Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, le potentiel de compression de champ est calculé suivant la formule $\sum_{i=1}^K N_i \frac{L_i}{H_i+a}$ dans
- 30 laquelle a est une constante strictement positive.

Ce mode de réalisation reprend la variante précédente mais en l'adaptant au cas où la section est de longueur variable.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, le potentiel de compression de champ est dit potentiel de compression corrigé et est

calculé pour une section donnée en multipliant le potentiel de compression de champ par un facteur de correction $\alpha(1 - \frac{K-\beta}{N+\gamma})$ dans lequel N est le nombre de fois que la section apparaît dans l'ensemble dit d'apprentissage ; K est le nombre de valeurs différentes que prend la section dans l'ensemble dit d'apprentissage ; et α , β et γ sont des coefficients positifs choisis.

- 5 Dans notre expérience, le potentiel de compression de champ permet de guider l'algorithme d'obtention mais présente le défaut suivant : quand une section présente des valeurs différentes réparties uniformément, l'estimateur d'entropie augmente trop lentement au fur et à mesure que de nouvelles valeurs sont observées dans l'ensemble dit d'apprentissage. Pour peu qu'elle ait une grande longueur, une section qui présente des valeurs à chaque fois différentes dans l'ensemble dit d'apprentissage pourra avoir un meilleur potentiel de compression de champ qu'une section avec peu de valeurs mais une taille plus petite. Or, à l'usage, c'est bien cette section avec peu de valeurs pour laquelle on souhaite créer une ou plusieurs règles de compression qui s'appliqueront au peu de valeurs observées. Alors que la section très longue avec des valeurs observées toutes différentes sera peu intéressante car il est probable qu'à l'usage, les valeurs qui apparaîtront dans cette section seront encore de nouvelles valeurs, et donc des valeurs qui ne seront pas comprimées par les règles de compression apprises sur un ensemble d'apprentissage où elles n'apparaissent pas.

- C'est pour corriger ce défaut que nous introduisons un facteur de correction qui permet d'obtenir un potentiel de compression corrigé. Les coefficients α , β et γ sont de préférence positifs pour assurer que le facteur de correction fasse évoluer le potentiel de compression dans le sens voulu. L'idée est qu'une section qui présente des valeurs toutes différentes doit être défavorisée par rapport à une section présentant une observation semblant moins aléatoire.

- Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, les coefficients positifs α , β et γ sont choisis pour prendre des valeurs constantes. Selon un mode de réalisation particulier de ce dernier mode, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, les coefficients positifs α , β et γ prennent les valeurs $\alpha = 1$, $\beta = 0$ et $\gamma = 0$.

- Grâce à ce dernier mode, le facteur de correction prend la valeur $1 - \frac{K}{N}$ et vaut donc zéro quand toutes les valeurs de la section sont différentes et que l'on a $K = N$. Une telle observation dans l'ensemble dit d'apprentissage d'une section dont toutes les valeurs sont différentes indique fortement que les valeurs en dehors de l'ensemble dit d'apprentissage seront également différentes, et qu'une telle section ne pourra pas être comprimée efficacement. Le facteur de correction va alors annuler le potentiel de compression de champ pour cette section. Quand une section prend une seule valeur, le facteur de correction vaut $1 - \frac{1}{N}$ et le potentiel de correction corrigé est donc diminué par rapport à la valeur initiale du potentiel de correction de champ, mais est finalement très peu diminué.

D'autres valeurs constantes peuvent bien sûr être prises pour affiner la correction apportée par le facteur de correction.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, les coefficients positifs α , β et γ prennent les valeurs $\alpha = 1$, $\beta = 1$ et $\gamma = 0$.

- 5 Grâce à ce dernier mode, le facteur de correction prend la valeur $1 - \frac{K-1}{N}$ et vaut donc exactement 1 quand la section prend une seule valeur, c'est-à-dire pour $K = 1$. A l'opposé, le facteur de correction ne s'annule pas quand $K = N$ ce qui peut être intéressant pour conserver un potentiel de compression de champ non nul même sur une section qui semble quand même être très aléatoire puisque toutes les valeurs prises par la section dans l'ensemble d'apprentissage sont différentes. Ainsi, le potentiel de compression de champ étant
- 10 non nul, l'algorithme d'obtention pourra dans certains contextes ajouter une ou des règles de compression relatives à la section.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, les coefficients positifs α , β et γ sont choisis en fonction de K et de N.

- 15 Ce mode de réalisation permet d'obtenir d'autres valeurs du facteur de correction pour affiner son choix en fonction des tailles des échantillons observés dans l'ensemble d'apprentissage. Par exemple, en prenant $\alpha = 1$, $\beta = 0,5K$ et $\gamma = N$, on obtient un facteur de correction qui vaut $1 - \frac{0,5K}{2N}$ et qui présente une influence différenciée selon la taille de l'échantillon des valeurs observées pour la section dont on calcule le potentiel de compression corrigé.

- 20 Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, l'un des potentiels de compression obtenu est dit potentiel de compression de valeur et est calculé pour une valeur V_i d'une section donnée suivant la formule $N(L - H_i)$ dans laquelle N est le nombre de fois que la valeur V_i apparaît dans l'ensemble dit d'apprentissage ; L est la longueur de la valeur V_i en bits ; et H_i est une grandeur dite estimateur d'entropie partielle d'une valeur qui est calculée suivant la formule $H_i = -f_i \log_2(f_i)$ dans laquelle f_i est la fréquence d'apparition de la valeur V_i .

- 25 Ce mode de réalisation construit, selon le même modèle que celui vu précédemment pour le potentiel de compression de champ, et en utilisant les mêmes grandeurs, un potentiel de compression adapté aux valeurs que prend une section donnée dans l'ensemble d'apprentissage, sachant qu'une section peut être un champ d'un en-tête de paquet ou une section délimitée d'une charge de paquet.

- 30 On peut noter que ce potentiel de compression de valeur s'exprime suivant la longueur de la valeur et est donc adapté à une section de longueur variable. Une valeur de faible longueur mais qui apparaît fréquemment pourra avoir un potentiel de compression supérieur à une valeur plus longue mais qui apparaît moins fréquemment. Ce potentiel va lui aussi guider la création de règles de compression en indiquant les valeurs qui sont à compresser en priorité. Ce ne sera donc pas forcément la valeur qui apparaît le plus

fréquemment puisque, dans le cas d'une section de longueur variable, la longueur effective de la valeur prise par la section est prise en compte et peut compenser une fréquence d'apparition plus faible.

Comme vu précédemment, la fréquence d'apparition d'une valeur est strictement positive et vaut 1 au maximum. L'estimateur d'entropie partielle d'une valeur sera donc également positif et vaudra 0 pour une valeur qui se trouverait être la valeur unique prise par une section dans l'ensemble dit d'apprentissage et qui aurait alors une fréquence d'apparition qui vaudrait 1. L'estimateur d'entropie partielle d'une valeur sera toujours strictement inférieur à 1. Le potentiel de compression d'une valeur sera donc toujours positif et sera maximal pour une valeur qui serait la valeur unique observée pour une section donnée et sera minimal pour la valeur qui est observée le moins souvent dans cette section dans l'ensemble dit d'apprentissage.

- 10 Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, le potentiel de compression de valeur est calculé suivant la formule $N \frac{L}{H_i + a}$ dans laquelle a est une constante strictement positive.

Grâce à ce mode de réalisation, une formule alternative est donnée pour calculer un potentiel de compression de valeur.

- 15 Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, l'un des potentiels de compression obtenu est dit potentiel de compression de bit de champ et est calculé pour un rang i d'une section donnée, les rangs i variant de 0 à $L - 1$, L étant la longueur minimale des valeurs prises par la section dans l'ensemble d'apprentissage, ladite longueur L étant exprimée en nombre de bits, et ladite grandeur est calculée suivant la formule $1 - H(b_i)$, dans laquelle $H(b_i)$ est une
- 20 grandeur dite estimateur d'entropie d'un bit de champ qui est calculée suivant la formule $H(b_i) = -(f_i \cdot \log_2(f_i) + (1 - f_i) \cdot \log_2(1 - f_i))$, ladite formule valant par convention 0 quand f_i vaut 0 ou 1, formule dans laquelle les valeurs prises par la section dans l'ensemble dit d'apprentissage sont exprimées en séquences de bits ordonnées et dans laquelle f_i est la fréquence d'apparition de l'une des valeurs 0 ou 1 pour le bit de rang i parmi les valeurs prises par la section dans l'ensemble dit d'apprentissage.
- 25 Ce mode de réalisation construit, selon les modèles vus précédemment, un potentiel de compression pour mesurer le potentiel de compression des bits d'une section dans un rang donné. Comme pour les potentiels déjà vus, cette grandeur est calculée pour une section qui peut être un champ d'en-tête ou une section délimitée de charge, et le terme « potentiel de compression de bit de champ » est retenu car il est utilisé, dans de nombreux modes de réalisation, pour s'appliquer à un champ d'en-tête.
- 30 Le principe est là encore de relever des fréquences d'apparition, ici les fréquences des valeurs de bits 0 ou 1 pour un rang donné de la section. Si cette fréquence est 1, cela veut dire que, pour un rang donné, c'est toujours la même valeur de bit qui apparaît, et l'estimateur d'entropie sera alors minimal et vaudra 0 par convention. Le potentiel de compression de bit de champ pour ce rang vaudra alors 1, et c'est la valeur maximale que peut prendre cette grandeur. Pour une fréquence de 0.5, le logarithme en base 2 vaut -1 et

l'estimateur d'entropie vaut 1, ce qui donne un potentiel de compression de bit de champ pour ce rang qui vaudra 0, ce qui est la valeur minimale que peut prendre cette grandeur.

Pour des fréquences de 0.25 et 0.75 pour un rang donné, les logarithmes en base 2 valent respectivement -2 et -0.415. L'estimateur d'entropie de bit de champ vaut alors $-(0,25 \cdot (-2) + 0,75 \cdot (-0.415)) = -(-0,5 - 0,31125) = 0,81125$ et le potentiel de compression de bit de champ vaut alors 0,18875, bien compris entre 0 et 1.

On voit encore comment le potentiel de compression, ici pour les rangs des sections dont les valeurs sont exprimées en suite de bits, va permettre de guider l'algorithme d'apprentissage en indiquant les valeurs les plus intéressantes à comprimer, en prenant ici en compte la représentation des valeurs sous forme de suite de bits.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, l'étape de sélection d'une règle appartenant à l'ensemble de règles de compression comprend la partition de l'ensemble dit d'apprentissage en sous-ensembles attribués à une règle de l'ensemble de règles de compression ; le calcul d'un potentiel de compression pour les règles de l'ensemble de règles de compression et la sélection de la règle selon son potentiel de compression, le potentiel de compression d'une règle étant calculé à partir de potentiels de compressions de champ calculés sur le sous-ensemble de l'ensemble dit d'apprentissage attribué à la règle de compression.

Grâce à ce mode de réalisation, les potentiels de compression de champ sont utilisés pour sélectionner une règle parmi l'ensemble de règles de compression qui est en cours d'obtention. L'ensemble dit d'apprentissage est partitionné en affectant à une règle donnée le sous-ensemble pour lequel la règle réalise la meilleure compression. Les potentiels de compression de champ sont calculés pour les champs non comprimés par la règle en question puis additionnés pour tous les paquets de l'ensemble dit d'apprentissage. De cette manière, toutes choses égales par ailleurs, l'obtention va chercher à améliorer une règle qui s'applique à plus d'entêtes qu'une autre et va également éviter de recréer des règles déjà créées.

Le point de départ de l'algorithme d'obtention étant un ensemble formé d'une seule règle qui s'applique à toutes les valeurs (en les ignorant) et qui ne fait pas de compression, cette règle sera sélectionnée initialement lors de la première itération de l'algorithme. Une règle sera ensuite créée à partir de cette règle initiale, qui va par exemple réaliser l'effacement d'une valeur qui survient fréquemment pour un champ. La règle créée sera donc celle qui réalise la meilleure compression pour les paquets présentant cette valeur, puisqu'elle l'efface, alors que la règle initiale ne fait rien. Le potentiel de compression de la règle est alors calculé en prenant en compte l'effacement réalisé pour cette valeur. La règle initiale, qui ne fait rien, pourra être sélectionnée ultérieurement car elle a un certain potentiel de compression pour les paquets qui ne présentent pas la valeur effacée par la règle créée.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, le calcul d'un potentiel de compression pour une règle s'effectue en additionnant les potentiels de compression de champs pour les sections restantes du sous-ensemble identifié.

Grâce à ce mode de réalisation, tous les potentiels de compression restant à exploiter sont pris en compte puisque c'est leur somme qui indique quelle est la règle qui va être la plus intéressante à sélectionner.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, le calcul d'un potentiel de compression pour une règle s'effectue en retenant le maximum des potentiels de compression de champs pour les sections restantes du sous-ensemble identifié.

Dans ce mode, plutôt que de réaliser une addition des potentiels de compression de champ, on retient la règle qui présente un maximum. Cette variante de calcul peut être plus efficace en opération de calcul, et plus rapide, tout en donnant un bon guidage de l'algorithme d'obtention.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, l'étape de sélection d'une règle appartenant à l'ensemble de règles de compression est réalisée par un tirage aléatoire d'une règle parmi l'ensemble de règles de compression.

Ce mode de réalisation donne une alternative au mode de réalisation qui consiste à calculer un potentiel de compression pour chaque règle. L'avantage de ce mode de réalisation est de proposer une alternative plus simple à réaliser.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, l'étape de création d'une nouvelle règle à partir de la règle sélectionnée s'effectue en créant une ou plusieurs règles candidates et en retenant une règle parmi les règles candidates créées selon l'efficacité de la compression réalisée par cette règle sur des paquets de l'ensemble d'apprentissage.

Grâce à ce dernier mode de réalisation, l'obtention est bien guidée par une fonction de récompense en lien avec la compression réalisée par l'ensemble de règles de compression qui est en cours d'obtention. Une règle est sélectionnée, à partir de laquelle une ou plusieurs règles candidates vont être créées, et c'est bien la règle qui réalise la compression la plus efficace qui est ajoutée à l'ensemble de règles de compression. Une possibilité est d'appliquer les règles de compression déjà existantes aux paquets de l'ensemble dit d'apprentissage avant d'évaluer la compression réalisée. Cela permet de faire progresser l'obtention de l'ensemble de règles de compression en étant guidé par une récompense d'amélioration de la compression, et en évitant de retomber sur la création des mêmes règles en appliquant les compressions de la règle sélectionnée avant d'évaluer la compression des règles candidates.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, l'étape de création d'une nouvelle règle à partir de la règle sélectionnée s'effectue en choisissant une section selon son potentiel de compression de champ et en créant des règles candidates formées en ajoutant à la règle sélectionnée une opération de compression s'appliquant aux valeurs prises par la section.

Grâce à ce mode de réalisation, un algorithme d'obtention est défini, qui permet de créer des règles de compression à partir d'une règle sélectionnée en étant guidé par les potentiels de compression. Une fois la règle sélectionnée, une ou plusieurs règles candidates sont créées, tout d'abord en sélectionnant une section

selon son potentiel de compression de champ puis en ajoutant à la règle sélectionnée une opération de compression qui s'applique aux valeurs prises par la section en question. Une opération de compression consiste à effacer une valeur complète ou une portion d'une section quand la section prend une valeur spécifiée dans l'opération. La valeur effacée pourra être remplacée par un index de dictionnaire. La connaissance de la règle de compression appliquée pourra permettre de reconstruire la ou les valeurs (ou portions de valeurs) effacées par la ou les opérations comprises dans la règle de compression appliquée. Ce sera la règle la plus efficace parmi les règles candidates qui sera retenue.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, une opération de compression ajoutée à la règle sélectionnée est une opération d'effacement de valeur, la valeur étant choisie selon les potentiels de compression de valeur des valeurs prises par la section choisie.

Grâce à ce mode de réalisation, une première règle candidate est créée, qui ajoute une opération très efficace de compression, à savoir l'effacement d'une valeur. La valeur est choisie grâce aux potentiels de compression de valeur, dont on a vu qu'ils prenaient en compte la longueur potentiellement variable des valeurs prises par la section.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, une opération de compression ajoutée à la règle sélectionnée est une opération de remplacement d'une valeur prise par la section choisie par un index d'un dictionnaire dans laquelle, si la règle sélectionnée comprend déjà, pour la section choisie, une opération de remplacement de valeur par un index d'un dictionnaire, la valeur dont le potentiel de compression de valeur est le plus élevé est ajoutée au dictionnaire ; sinon, un dictionnaire est créé comprenant les deux valeurs dont le potentiel de compression de valeur est le plus élevé.

Grâce à ce mode de réalisation, une autre espèce de règle candidate est créée, ou améliorée, qui consiste à remplacer une valeur par un index de dictionnaire. Alors que l'opération d'effacement d'une valeur est efficace dans un contexte où une section prend très fréquemment une valeur donnée, l'opération de remplacement par un index de dictionnaire peut être plus efficace pour des valeurs dont la fréquence d'apparition est plus répartie.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, une opération de compression ajoutée à la règle sélectionnée est une opération d'effacement de préfixe jusqu'à un rang i , le rang i étant déterminé en multipliant entre eux les potentiels de compression de bits de champ de la section choisie en partant du rang 0 jusqu'au rang maximum, jusqu'à ce que le produit soit inférieur à un seuil donné, par exemple dès que le produit devient strictement inférieur à 1.

Grâce à ce mode de réalisation, une autre espèce de règle candidate est créée qui consiste à effacer les préfixes des valeurs prises par une section donnée. Les potentiels de compression de bits de champ permettent de détecter les préfixes communs à de nombreuses valeurs.

Grâce à l'ensemble de ces modes de réalisation, un algorithme d'obtention est défini, qui permet de créer des règles de compression à partir d'une règle sélectionnée en étant guidé par les potentiels de compression. L'algorithme peut créer à chaque étape des règles candidates, chaque règle pouvant correspondre à une opération de compression parmi les opérations principales définies par le protocole SCHC.

- 5 Pour créer les règles candidates, l'algorithme d'obtention commence par sélectionner la section dont le potentiel de compression de champ est le plus fort. Les règles candidates créées vont s'appliquer à une ou des valeurs de cette section.

10 Une première règle candidate consiste en un ajout d'une opération d'effacement de valeur à la règle sélectionnée. La valeur effacée est tout simplement celle dont le potentiel de compression de valeur est le plus élevé, ce qui n'est pas forcément la valeur la plus fréquente dans le cas d'une section à longueur variable.

15 Une deuxième règle consiste en un ajout d'une opération de remplacement d'une valeur par un index d'un dictionnaire à la règle sélectionnée. Si la règle sélectionnée comprend déjà une opération de remplacement par un index de dictionnaire pour cette section, la modification va consister en l'ajout d'une nouvelle référence au dictionnaire pour la valeur dont le potentiel de compression est le plus élevé. Sinon, la modification va créer une opération de type remplacement par un index de dictionnaire ; et comme cette opération n'a de sens que pour au moins deux valeurs, le dictionnaire créé va comprendre les deux valeurs dont les potentiels de compression de valeur sont les plus élevés.

20 Enfin, une troisième règle consiste en un ajout d'une opération d'effacement d'un préfixe de bits. Pour définir la règle, il est important de décider jusqu'à quel rang effacer le préfixe de bits. Pour cela, on utilise les potentiels de compression de bits de champ. Si un potentiel de compression de bit vaut 1 pour un rang donné, cela veut dire que les valeurs pour ce rang ont toutes le même bit 0 ou 1. En partant du premier rang, et en multipliant les potentiels de compression de bits les uns avec les autres, on va décroître et l'algorithme consiste en choisir de s'arrêter quand un seuil donné est franchi par le produit. Si les valeurs pour les deux premiers rang, par exemple, ont toutes le même bit, le produit des potentiels de compression de bits vaudra 1. Si le rang suivant a une fréquence 0.5 pour les bits 0 et 1, le potentiel de compression de bit pour ce rang vaut zéro, et le produit s'annulera. Le passage du produit sous un seuil est donc évident, et la règle créée effacera bien le préfixe pour les deux premiers rangs et pas pour le troisième. En choisissant que le produit doit rester au minimum égal à un, l'algorithme ne va créer des règles candidates d'effacement de préfixe que pour des préfixes qui prennent bien une même valeur exprimée en suites de bits jusqu'au rang pour lequel le produit vaut moins que 1.

35 L'avantage de ce dernier mode de réalisation, comparé à un mode de réalisation où, par exemple, les règles candidates seraient créées au hasard, est de s'assurer que les potentiels de compression guident bien la création de nouvelles règles. De cette façon, l'ensemble de règles de compression sera de taille limitée car les règles de compression les plus efficaces seront créées en premier pour être ajoutées à l'ensemble de règles de compression.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, un ordre prédéfini est appliqué permettant de choisir la règle candidate à ajouter en cas d'égalité entre les compressions réalisées par des règles candidates.

- 5 Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, l'ordre prédéfini permettant de choisir la règle candidate consiste à favoriser la règle candidate réalisant un effacement de valeur, puis la règle candidate réalisant un remplacement par un index de dictionnaire, puis la règle candidate réalisant un effacement de préfixe.

Grâce à ces modes de réalisation, un ordre prédéfini permet de s'assurer qu'un choix sera bien fait entre les règles candidates en cas d'égalité de la compression réalisée par ces règles.

- 10 Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, les règles candidates sont créées en ajoutant ou modifiant de façon aléatoire une opération de compression à la règle sélectionnée.

- Grâce à ce mode, une variante de l'algorithme est proposée qui n'utilise pas le guidage par les potentiels de compression. Un algorithme d'obtention peut également fonctionner en utilisant des changements d'état au hasard, et c'est la fonction de récompense qui va permettre de s'assurer que l'obtention progresse bien vers un résultat acceptable. L'avantage de ce mode de réalisation est de proposer une variante plus simple à mettre en œuvre.
- 15

- Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, l'ensemble dit d'apprentissage est séparé en plusieurs sous-ensembles de paquets de données, les paquets regroupés dans un sous-ensemble ayant tous un même format d'en-tête correspondant à un empilement de protocoles, et caractérisé en ce que l'obtention d'un ensemble de règles de compression est appliquée séparément aux sous-ensembles de l'ensemble dit d'apprentissage avant de regrouper les ensembles de règles obtenus pour les sous-ensembles en un seul ensemble de règles.
- 20

- Grâce à ce mode de réalisation, l'algorithme d'obtention peut être utilisé sur un ensemble dit d'apprentissage formé de plusieurs sous-ensembles distincts, correspondant chacun à des paquets ayant un format donné d'en-têtes, ce format correspondant lui-même à un empilement donné de protocoles. L'obtention se fait alors pour un format donné d'en-têtes, puis les différents ensembles de règles de compression sont réunis en un seul. On obtient ainsi un moyen efficace de couvrir un contexte de déploiement des règles de compression s'appliquant aux paquets de plusieurs protocoles.
- 25

- 30 Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les modes précédents, les règles de compression de l'ensemble de règles de compression obtenu et les opérations de compression comprises dans lesdites règles de compression sont des règles et des opérations de compression telles que définies dans le protocole SCHC.

Grâce à ce mode de réalisation, l'objectif de proposer une technique d'apprentissage automatique de règles de compression pour le protocole SCHC est bien tenu. Le problème technique de disposer d'un moyen de déploiement du protocole SCHC sans solliciter d'expert pour définir les règles de compression est bien résolu.

Selon un autre aspect fonctionnel, l'invention a trait à un procédé de gestion de la transmission d'un paquet de données dans un réseau, le paquet comprenant des sections prenant des valeurs respectives, caractérisé en ce qu'il comprend les étapes suivantes :

- l'obtention préalable d'un ensemble de règles de compression comprenant des opérations de compression qui peuvent s'appliquer aux valeurs prises par les sections dudit paquet de données ;
- l'application d'au moins une opération de compression à une valeur de section dudit paquet de données, ladite opération étant comprise dans une règle de compression appartenant audit ensemble de règles de compression obtenu ;
- la transmission dudit paquet de données et d'une donnée représentative de la règle de compression à laquelle appartient ladite opération ;
- la réception dudit paquet de données et de ladite donnée représentative ;
- l'application audit paquet de données d'au moins une opération de décompression symétrique de l'au moins une opération de compression appliquée à une valeur de section dudit paquet de données.

Grâce à cet aspect de l'invention, une transmission complète de paquet de données est réalisée, en transmettant le paquet de données compressé. L'obtention des règles de compression permet de rendre le procédé automatique, sans intervention d'expert. La transmission de la donnée représentative de la règle de compression qui a été appliquée au paquet transmis permet de retrouver cette règle et donc les opérations de compression comprises dans la règle, et de rétablir le paquet original en appliquant les opérations de décompression symétriques.

On a donc transmis un paquet compressé au lieu du paquet original, ce qui permet de très fortes économies en termes de ressources matérielles et énergétiques. Comme les règles de compression sont obtenues automatiquement, il n'y a pas besoin de faire intervenir un expert, ce qui permet de déployer très facilement la transmission efficace selon l'invention.

Selon un mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec le mode précédent, le procédé comprend de surcroît une étape préalable de collecte dans le réseau de paquets transmis pour former un ensemble dit d'apprentissage utilisé lors de l'étape d'obtention d'un ensemble de règles de compression et une étape de déploiement synchronisé dans les nœuds du réseau de l'ensemble de règles de compression obtenu.

Grâce à ce mode de réalisation, la transmission de paquets dans le réseau se fait en prenant en compte les paquets effectivement transmis.

L'obtention préalable de l'ensemble de règles de compression utilise un général un ensemble dit d'apprentissage. Celui-ci peut être fourni par un expert, ou bien généré automatiquement selon des

contraintes prédéfinies. Mais le mode de réalisation préféré est celui où l'ensemble dit d'apprentissage est collecté parmi les paquets transmis dans le réseau. L'obtention se fait alors avec un ensemble dit d'apprentissage directement représentatif du trafic de paquets dans le réseau et les règles de compression obtenues seront donc bien plus adaptées. Une fois l'ensemble de règles de compression obtenu, il doit être

5 déployé dans les nœuds du réseau de manière synchronisée afin qu'un nœud qui reçoit un paquet compressé, et la donnée représentative de la règle de compression utilisée, puisse retrouver cette règle et appliquer l'opération de décompression symétrique.

De plus, grâce à ce mode de réalisation, le procédé peut être mis en œuvre de façon permanente. La collecte de paquets de données servant à la formation de l'ensemble dit d'apprentissage est permanente et permet de

10 capturer des changements de contexte dans la transmission en cours de paquets. Un premier ensemble de règles de compression peut être déjà déployé et, si le contexte change, par exemple si les adresses IP des nœuds du réseau changent, les règles de compression deviennent inefficaces. Mais les paquets collectés permettent de relancer l'obtention des règles de compression avec un nouvel ensemble dit d'apprentissage, et le déploiement d'un nouvel ensemble de règles de compression permet alors de s'adapter au nouveau

15 contexte, et de réaliser de nouveau des compressions efficaces. Le déploiement doit être synchronisé afin de s'assurer que le nœud qui reçoit un paquet compressé connaît bien la règle de compression qui a été appliquée afin de pouvoir effectuer l'opération de décompression symétrique pour rétablir le paquet de données.

Selon un autre mode de réalisation, qui pourra être mis en œuvre alternativement ou cumulativement avec les

20 modes précédents, les règles de compression de l'ensemble de règles de compression obtenu, les opérations de compression comprises dans lesdites règles de compression, et les opérations symétriques de décompression sont des règles et des opérations de compression et de décompression telles que définies dans le protocole SCHC.

Grâce à ce mode de réalisation, le protocole SCHC peut être déployé dans des contextes dynamiques. Les

25 ensembles de règles de compression seront mis à jour au fur et à mesure de l'obtention utilisant des nouveaux ensembles dit d'apprentissage représentatifs du trafic entre les différents points d'un réseau, et des changements dans les en-têtes de paquets circulant effectivement. Le protocole SCHC est donc rendu apte à s'appliquer à des contextes dynamiques grâce à l'utilisation de l'apprentissage par renforcement pour créer les ensembles de règles de compression.

Selon un premier aspect matériel, l'invention a trait à une entité de gestion de l'obtention d'un ensemble de

30 règles de compression aptes à compresser un paquet de données transmis dans un réseau, le paquet comprenant des sections prenant des valeurs respectives exprimées sous la forme de suite de bits, le paquet étant apte à être compressé par application de règles de compression aptes à supprimer des bits d'une valeur prise par une section du paquet, caractérisée en ce que l'entité de gestion comprend les modules suivants :

- 35
- un module d'obtention d'un potentiel de compression pour une section donnée, relativement aux valeurs prises par la section donnée dans les paquets appartenant à un ensemble de paquets de données, ensemble dit d'apprentissage ;

- un module d'obtention de l'ensemble de règles de compression en fonction des potentiels de compression obtenus.

Selon un autre aspect matériel, l'invention a trait à un paquet de données formé en concaténant une donnée représentative d'une règle de compression, ladite donnée étant un identifiant de la règle de compression consistant en une suite de bits, avec un paquet de données auquel a été appliqué au moins une opération de compression appartenant à ladite règle de compression qui appartient à un ensemble de règles de compression obtenu par le procédé.

Selon un autre aspect matériel, l'invention a trait à un équipement réseau comprenant une entité de gestion de l'obtention d'un ensemble de règles de compression selon l'invention.

Selon un autre aspect matériel, l'invention a trait à un système de gestion de la transmission d'un paquet de données dans un réseau, le paquet comprenant des sections prenant des valeurs respectives, caractérisé en ce que le système comprend les modules suivants :

- un module d'obtention préalable d'un ensemble de règles de compression comprenant des opérations de compression qui peuvent s'appliquer aux valeurs prises par les sections d'un paquet de données ;
- un module d'application d'au moins une opération de compression à une valeur de section d'un paquet de données, ladite opération étant comprise dans une règle de compression appartenant à un ensemble de règles de compression obtenu ;
- un module de transmission d'un paquet de données auquel a été appliqué une opération de compression et d'une donnée représentative d'une règle de compression à laquelle appartient ladite opération de compression ;
- un module de réception d'un paquet de données et d'une donnée représentative d'une règle de compression ;
- un module d'application à un paquet de données d'une opération de décompression symétrique d'une opération de compression appliquée à une valeur de section dudit paquet de données.

Selon un mode de réalisation, le système de gestion de la transmission comprend de surcroît :

- un module de collecte dans le réseau de paquets de données transmis pour former un ensemble dit d'apprentissage utilisé par un module d'obtention d'un ensemble de règles de compression ;
- un module de déploiement synchronisé dans les nœuds du réseau d'un ensemble de règles de compression obtenu.

Enfin, selon un autre aspect matériel, l'invention a trait à des supports de données sur lesquels sont enregistrés des programmes d'ordinateurs comprenant des séquences d'instructions pour la mise en œuvre des procédés définis ci-dessus.

Les supports de données peuvent être n'importe quelle entité ou dispositif capable de stocker les programmes. Par exemple, les supports peuvent comporter un moyen de stockage, tel qu'une ROM, par exemple un CD ROM ou une ROM de circuit microélectronique, ou encore un moyen d'enregistrement magnétique tel qu'un disque dur. D'autre part, les supports peuvent être des supports transmissibles tels qu'un signal électrique ou optique, qui peuvent être acheminés via un câble électrique ou optique, par radio ou par d'autres moyens. Les programmes selon l'invention peuvent être en particulier téléchargés sur un réseau de type Internet. Alternativement, le support d'informations peut être un circuit intégré dans lequel le programme est incorporé, le circuit étant adapté pour exécuter ou pour être utilisé dans l'exécution du procédé en question.

Brève description des figures

L'invention sera mieux comprise à la lecture de la description qui suit, donnée à titre d'exemple, et faite en référence aux dessins annexés sur lesquels :

[Fig 1] représente une entité d'obtention d'un ensemble de règles de compression ainsi qu'une entité de gestion de la compression d'un paquet de données transmis dans un réseau et une entité de gestion de la décompression d'un paquet de données.

[Fig 2] représente un exemple d'obtention d'un ensemble de règles de compression par la mise en œuvre d'un algorithme d'apprentissage. La figure 2 est subdivisée en quatre sous-figures 2.A, 2.B, 2.C et 2.D, correspondant à quatre étapes d'apprentissage.

[Fig 3] représente un exemple de calcul de potentiels de compression de champ explicitant l'intérêt d'utiliser un potentiel de compression de champ corrigé.

[Fig 4] représente un exemple de calcul de potentiels de compression de valeurs explicitant la différence entre un potentiel de compression d'une valeur et sa fréquence d'apparition.

[Fig 5] représente un exemple de système de transmission de paquets de données selon l'invention, montrant le déploiement d'un tel système dans un réseau comprenant une passerelle et deux terminaux.

[Fig 6] représente un autre exemple de système de transmission de paquets de données selon l'invention, montrant le déploiement d'un tel système dans un réseau comprenant deux routeurs.

Description détaillée

La figure 1 représente une entité 300 d'obtention d'un ensemble RS de règles de compression ainsi qu'une entité de gestion de la compression 100 d'un paquet de données P transmis dans un réseau NET et une entité de gestion de la décompression 200 d'un paquet de données P.

L'entité 300 d'obtention d'un ensemble RS de règles de compression comprend un module 301 d'obtention d'un potentiel de compression PC relativement aux valeurs prises par les sections de paquets de données appartenant à un ensemble E de paquets de données, ensemble dit d'apprentissage ainsi qu'un module APP d'obtention de l'ensemble RS de règles de compression en fonctions des potentiels de compression PC obtenus.

Dans des exemples de réalisation, le module APP met en œuvre un algorithme d'apprentissage afin d'obtenir l'ensemble RS de règles de compression. Les exemples présentés dans la figure 1 et les figures suivantes correspondent à ces exemples de réalisations dans lesquels un algorithme d'apprentissage est mis en œuvre. Dans ces exemples, l'algorithme d'apprentissage utilise un ensemble d'apprentissage qui est l'ensemble dit d'apprentissage.

En plus de l'entité d'obtention 300, la figure 1 décrit deux entités de gestion de la compression 100 et de la décompression 200 d'un paquet P de données afin d'illustrer l'utilisation de l'ensemble RS de règles de compression obtenu par la mise en œuvre du procédé par l'entité d'obtention 300.

Les deux entités de gestion de la compression 100 et de la décompression 200 appartiennent à un même réseau NET qui est un réseau de communication utilisant des protocoles de transmission de paquets de données P. Dans l'exemple de réalisation décrit dans la figure 1, l'entité d'obtention 300 appartient au même réseau NET mais dans d'autres exemples de réalisation, l'entité d'obtention 300 peut appartenir à des réseaux différents ou même être isolée du réseau NET. L'ensemble RS de règles de compression obtenu par l'entité d'obtention 300 pourra être déployé dans les entités de gestion de la compression 100 et de la décompression 200 par tout moyen. Le paquet de données P comprend des sections prenant des valeurs respectives. Les deux entités de gestion de la compression 100 et de la décompression 200 disposent d'un même ensemble de règles de compression RS, cet ensemble comprenant des règles de compression qui comprennent elles-mêmes des opérations de compression qui s'appliquent à des valeurs prises par les sections d'un paquet de données P. L'ensemble RS de règles de compression a été obtenu par la mise en œuvre d'un algorithme d'obtention par l'entité 300.

L'entité de gestion de la compression 100 comprend un module 101 d'application d'au moins une opération de compression à une valeur de section d'un paquet de données P, ladite opération étant comprise dans une règle de compression appartenant à un ensemble de règles de compression RS, ledit ensemble RS étant obtenu par apprentissage.

L'entité de gestion de la compression 100 comprend un module 102 de transmission d'un paquet de données P auquel au moins une opération de compression a été appliquée et d'une donnée représentative de la règle de compression à laquelle appartient ladite opération.

L'entité de gestion de la décompression 200 comprend un module 201 de réception d'un paquet de données P auquel au moins une opération de compression a été appliquée à une valeur de section dudit paquet P, ladite opération étant comprise dans une règle de compression appartenant à un ensemble de règles de

compression RS, ledit ensemble RS étant obtenu par apprentissage, et d'une donnée représentative de la règle de compression à laquelle appartient ladite opération.

L'entité de gestion de la décompression 200 comprend un module 202 d'application à un paquet de données P d'au moins une opération de décompression symétrique d'une opération de compression appliquée à une valeur de section dudit paquet P.

La transmission du paquet de données P par le module 102, et la réception du même paquet P par le module 201, est représentée par une flèche reliant le module 102 au module 201, étiquetée par le paquet P. La transmission et la réception de la donnée représentative de la règle de compression appliquée au paquet P ne sont pas représentées dans la figure 1. Dans certains modes de réalisation, le paquet de données P et la donnée représentative sont transmis et reçus conjointement.

Comme l'entité de gestion de la compression 100 et de la décompression 200 disposent du même ensemble de règles de compression RS obtenu, l'entité de gestion de la décompression 200 pourra utiliser la donnée représentative de la règle de compression appliquée au paquet P pour retrouver celle-ci. Il est aisé ensuite de déterminer l'opération symétrique de décompression. Le module 202 d'application d'une opération de décompression symétrique permet alors de rétablir le paquet de données P tel qu'il était avant l'application d'une opération de compression.

Le paquet de données P a donc été transmis dans le réseau après application sur les valeurs prises par les sections du paquet P d'une ou plusieurs opérations de compression par l'entité de gestion de la compression 100. La transmission du paquet P une fois compressé permet de réaliser d'importantes économies de ressources énergétiques et matérielles. Le paquet P est ensuite rétabli par l'entité de gestion de la décompression 200 pour utilisation ultérieure. Comme l'ensemble de règles de compression RS est obtenu par apprentissage, le procédé ne demande pas d'intervention d'un expert, et il peut s'adapter à des contextes changeant en réitérant l'algorithme d'apprentissage pour disposer d'ensembles de règles de compression RS adaptés au contexte.

La figure 2, quant à elle, représente un exemple d'exécution d'un mode de réalisation d'obtention d'un ensemble de règles de compression selon un mode de réalisation du procédé de l'invention. Dans cet exemple, un algorithme d'apprentissage est utilisé. La figure 2 est subdivisée en quatre sous-figures (2.A, 2.B, 2.C et 2.D) qui correspondent à quatre étapes S0, S1, S2 et S2' de l'apprentissage selon ce mode de réalisation.

La figure 2.A correspond à l'état initial S0 de l'apprentissage et à la première étape.

L'ensemble d'apprentissage E est constitué de N=10000 paquets. Ces N=10000 paquets contiennent un en-tête formé, dans notre exemple, de quatre champs C1, C2, C3, C4. Dans notre exemple, chacun des champs C1, C2, C3, C4 a une longueur de 8 bits, indiquée par la mention L=8.

Pour le premier champ C1, les $N=10000$ paquets de l'ensemble d'apprentissage E présentent tous la même valeur V_1^1 non spécifiée. La fréquence f_1^1 d'apparition de la valeur V_1^1 est donc de 100%.

Pour le deuxième champ C2, les $N=10000$ paquets de l'ensemble d'apprentissage E présentent trois valeurs V_2^1 , V_2^2 et V_2^3 qui sont là encore non spécifiées. Seules les fréquences d'apparition f_2^1 , f_2^2 et f_2^3 de ces trois valeurs sont précisées. Elles valent respectivement 80%, 10% et 10%.

Pour le troisième champ C3, les $N=10000$ paquets de l'ensemble d'apprentissage présentent quatre valeurs V_3^1 , V_3^2 , V_3^3 et V_3^4 qui apparaissent dans l'ensemble d'apprentissage avec la même fréquence de 25%. Les valeurs sont spécifiées comme suite de 8 bits, et on voit qu'elles se différencient par le préfixe.

Pour le quatrième champ C4, les $N=10000$ paquets de l'ensemble d'apprentissage présentent quatre valeurs V_4^1 , V_4^2 , V_4^3 et V_4^4 qui apparaissent là aussi dans l'ensemble d'apprentissage avec la même fréquence de 25%. Les valeurs sont spécifiées comme suite de 8 bits, mais ici, contrairement aux valeurs du champ C4, ces quatre valeurs ont un préfixe identique sur les 6 premiers bits.

Cet exemple est bien sûr fictif, et choisi pour illustrer le fonctionnement de l'algorithme d'apprentissage de règles de compression. Les différentes fréquences d'apparition et valeurs prises vont permettre d'illustrer les différentes opérations de compression pouvant être ajoutées aux règles de compression lors de l'apprentissage selon un mode de réalisation. Dans l'exemple ici présenté, la compression va chercher uniquement à compresser les valeurs prises par les champs C1, C2, C3 et C4 des en-têtes des paquets de données et pas des sections délimitées des charges des paquets, mais les principes resteraient les mêmes dans le cas où l'on chercherait à apprendre des règles de compression qui s'appliquent à des sections des charges des paquets.

On suppose que les fréquences d'apparition des différentes valeurs sont indépendantes les unes des autres. Par exemple, on suppose que dans les 8000 paquets de l'ensemble d'apprentissage E pour lesquels le champ C2 prend la valeur V_2^1 , les quatre valeurs des champs C3 et C4 apparaissent de façon équiprobable, comme dans l'ensemble d'apprentissage E complet. Ces valeurs apparaissent donc respectivement chacune dans 2000 paquets des 8000 paquets où le champ C2 prend la valeur V_2^1 . On suppose que cette propriété est également vraie en combinant les différentes valeurs de champ. Par exemple, dans les 2000 paquets pour lesquels à la fois le champ C2 prend la valeur V_2^1 et le champ C3 prend la valeur V_3^1 , alors les fréquences d'apparition des valeurs du champ C4 sont également équiprobables, et on trouve donc 500 paquets pour chacune de ces quatre valeurs. Cette hypothèse nous permettra de simplifier la présentation d'un mode de réalisation de l'algorithme sans perdre en généralité.

Les fréquences d'apparition sont présentées en pourcentage dans la figure 2.A et suivantes, mais, pour les calculs à suivre, les fréquences sont des grandeurs strictement positives, inférieures ou égales à 1. Un pourcentage de 100% correspond à une grandeur de 1 ; un pourcentage de 80% à 0.8 ; un pourcentage de 25% à 0.25 et un pourcentage de 10% à 0.1.

Il est possible de calculer des potentiels de compression de champ PCC pour les quatre champs C1, C2, C3 et C4.

Selon un mode de réalisation de l'invention, le potentiel de compression de champ PCC se calcule, pour une section donnée, selon la formule $N(L - H)$, dans laquelle N est le nombre de fois que la section apparaît dans l'ensemble d'apprentissage ; L est la longueur de la section exprimée en bits et H est une grandeur dite estimateur d'entropie de champ qui est calculée suivant la formule $H = -\sum_{i \in [1, K]} f_i \log_2(f_i)$ dans laquelle K est le nombre de valeurs différentes $V_{i \in [1, K]}$ que prend la section dans l'ensemble d'apprentissage et f_i est la fréquence d'apparition de la valeur V_i .

On utilise, dans ce mode de réalisation et dans cet exemple, la formule dans laquelle la section, ici un des quatre champs C1, C2, C3 ou C4, est de longueur fixe. C'est bien le cas dans cet exemple, avec une longueur L égale à 8 bits pour les quatre champs C1, C2, C3, C4.

Pour le champ C1, le calcul donne donc le résultat suivant :

$$PCC(C1) = 10000 * (8 - (-f_1^1 \log_2(f_1^1))) = 10000 * (8 - (-1 * 0)) = 80000$$

Pour le champ C2, le calcul donne donc le résultat suivant :

$$\begin{aligned} PCC(C2) &= 10000 * (8 - (-f_2^1 \log_2(f_2^1) - f_2^2 \log_2(f_2^2) - f_2^3 \log_2(f_2^3))) \\ &= 10000 * (8 - (0.8 * 0.3219 + 0.1 * 3.322 + 0.1 * 3.322)) = 10000 * (8 - 0.92192) = 70781 \end{aligned}$$

Pour les champs C3 et C4, le calcul donne le même résultat car les fréquences d'apparition des valeurs sont les mêmes dans les deux champs. Le résultat est donné ci-dessous :

$$\begin{aligned} PCC(C3) &= 10000 * (8 - (-f_3^1 \log_2(f_3^1) - f_3^2 \log_2(f_3^2) - f_3^3 \log_2(f_3^3) - f_3^4 \log_2(f_3^4))) \\ &= 10000 * (8 - (4 * 0.25 * 2)) = 60000 = PCC(C4) \end{aligned}$$

A l'étape S0, dans le mode de réalisation présenté ici, l'ensemble de règles de compression RS est initialisé avec une règle unique R0. Cette règle est une règle qui correspond à toutes les valeurs possibles de tous les champs des paquets de l'ensemble d'apprentissage E, mais n'applique aucune compression à ces valeurs.

Ces éléments correspondent à l'état initial de l'algorithme d'apprentissage. Nous allons maintenant détailler la première étape de l'algorithme d'apprentissage par renforcement de l'ensemble de règles de compression RS, pour un mode de réalisation parmi plusieurs. Cette première étape est également représentée sur la figure 2.A.

L'algorithme d'apprentissage par renforcement de l'ensemble de règles de compression RS commence par la sélection d'une règle parmi l'ensemble RS. Comme l'ensemble RS comprend seulement la règle R0 à ce stade, c'est cette règle R0 qui va être sélectionnée. On verra dans les étapes ultérieures comment la sélection peut être effectuée lorsqu'il y a plusieurs règles dans l'ensemble de règles de compression RS.

A partir de la règle R0, plusieurs règles candidates vont être créées dans l'exemple de mode de réalisation présenté ici. La première étape pour créer les règles candidates consiste dans notre exemple à choisir un champ parmi les champs C1, C2, C3, C4. Pour cela, le mode de réalisation de l'algorithme d'apprentissage utilise les potentiels de compression de champ PCC(C1), PCC(C2), PCC(C3) et PCC(C4). En général, l'algorithme va ici choisir une section en utilisant les potentiels de compression de champ pour l'ensemble des sections dont on cherche à compresser les valeurs prises.

Ce calcul des potentiels de compression de champ PCC doit prendre en compte les compressions effectuées par les règles de compression présentes dans l'ensemble de règles de compression RS. Pour cela, un mode de réalisation possible est de réaliser une partition de l'ensemble d'apprentissage E selon les règles de compression de l'ensemble RS, et de calculer les potentiels de compression de champ PCC pour chaque règle de l'ensemble RS en utilisant les sous-ensembles de l'ensemble d'apprentissage E attribués à chaque règle. A ce stade initial, seule la règle R0, qui ne réalise aucune compression est présente. Le calcul des potentiels de compression de champ PCC est donc celui qui a été présenté ci-dessus.

Comme le potentiel de compression de champ PCC(C1) est le plus grand parmi les quatre potentiels PCC(C1), PCC(C2), PCC(C3) et PCC(C4), c'est ce champ C1 qui sera choisi pour créer les règles candidates à partir de la règle sélectionnée R0.

Dans notre exemple de réalisation, trois règles candidates vont être créées, correspondant à l'ajout à la règle sélectionnée R0 de trois opérations de compression possibles sur les valeurs prises par le champ C1 dans les paquets de l'ensemble d'apprentissage E.

La première règle candidate créée consiste en l'ajout d'une opération d'effacement. Pour sélectionner la valeur à effacer, dans notre exemple de réalisation, des potentiels de compression de valeur PCV sont calculés. En l'occurrence, comme le champ C1 prend une seule valeur V_1^1 dans l'ensemble d'apprentissage E, c'est bien sûr cette valeur qui sera effacée.

La deuxième règle candidate créée consiste en l'ajout d'une opération de remplacement par un index d'un dictionnaire. Dans ce cas, comme il y a une seule valeur prise par le champ C1, l'opération ajoutée consiste en remplacer cette valeur unique V_1^1 par un index de dictionnaire. Comme le dictionnaire en question serait utilisé pour une seule valeur, il ne comprendrait qu'un seul bit.

La troisième règle candidate créée consiste en l'ajout d'une opération d'effacement des préfixes communs aux valeurs prises par le champ. Ici, comme il y a une seule valeur prise par le champ C1, l'opération ajoutée consiste en l'effacement de la valeur elle-même, puisque tous les bits de préfixe sont évidemment communs pour cette valeur unique.

L'étape suivante de l'algorithme d'apprentissage va consister en retenir une règle parmi les règles candidates créées pour l'ajouter à l'ensemble de règles de compression RS. Pour cela, la fonction de récompense utilisée dans notre mode de réalisation est la compression réalisée par les règles candidates sur les paquets de l'ensemble d'apprentissage E.

La première règle candidate réaliserait l'effacement de la valeur V_1^1 prise par le champ C1. Comme ce champ a une longueur de 8 bits, et que la valeur V_1^1 apparaît dans les 10000 paquets de l'ensemble d'apprentissage E (sa fréquence d'apparition f_1^1 vaut 100%), la compression réalisée par cette règle candidate est de 8×10000 , soit 80000 bits.

- 5 La deuxième règle candidate réaliserait le remplacement de la valeur V_1^1 prise par le champ C1 par un index d'un dictionnaire. Comme cet index a une longueur d'un bit, la compression réalisée par cette règle pour un paquet ne serait pas de 8 bits, mais seulement de 7 bits. En multipliant par les 10000 paquets de l'ensemble d'apprentissage, la compression réalisée par cette règle candidate est de 70000 bits.

La troisième règle candidate réaliserait l'effacement du préfixe commun aux valeurs prises par le champ C1.

- 10 Comme le champ C1, dans l'ensemble d'apprentissage E, prend uniquement la valeur V_1^1 , alors toute cette valeur serait effacée par l'opération d'effacement du préfixe commun. Cette troisième règle candidate réalise donc la même compression que la première règle candidate, à savoir 80000 bits.

Dans notre mode de réalisation, la règle candidate choisie est celle qui réalise la meilleure compression. En cas d'égalité, un critère de choix peut être défini. Dans notre mode de réalisation, l'opération d'effacement est

15 privilégiée devant l'opération de remplacement par un index, elle-même devant l'opération d'effacement de préfixe commun. C'est donc cette règle de compression R1, notée $R1: [ns(V_1^1)]$, qui va être rajoutée à l'ensemble RS de règles de compression. La notation $ns(V_1^1)$ utilise l'acronyme ns pour l'anglais *not sent* pour indiquer une valeur non transmise. Une règle de compression est notée dans nos figures comme une suite d'opérations placées entre crochets.

- 20 A ce stade de l'algorithme d'apprentissage, l'ensemble de règles de compression RS comprend donc deux règles, la règle initiale R0 et la règle R1 qui vient de lui être ajoutée.

Dans le mode de réalisation présenté ici, lorsqu'une règle ajoutée s'applique à tous les paquets de données de l'ensemble d'apprentissage auxquels la règle sélectionnée s'applique, alors la règle sélectionnée est retirée de l'ensemble de règles de compression. C'est bien le cas ici : la règle initiale R0, qui ne réalise

25 aucune compression, s'applique bien à tous les paquets de l'ensemble d'apprentissage E et c'est aussi le cas de la règle R1 qui efface la valeur V_1^1 , puisque cette valeur est prise par 100% des champs C1 des paquets de l'ensemble d'apprentissage E.

La règle initiale R0 est donc retirée de l'ensemble RS, ce qui est représentée dans la figure 2.A par la case barrée placée sur elle.

- 30 Nous discuterons plus loin des critères d'arrêt de l'apprentissage par renforcement dans notre mode de réalisation, et nous supposons que l'algorithme d'apprentissage va bien se poursuivre. Nous allons donc voir maintenant l'exécution de l'étape suivante S1 de l'apprentissage par renforcement dans notre mode de réalisation, représentée par la sous-figure 2.B.

La figure 2.B correspond à l'état S1 de l'apprentissage, après la première étape, et à la deuxième étape de

35 l'apprentissage par renforcement selon un mode de réalisation.

L'ensemble d'apprentissage E n'est pas modifié, et comprend toujours 10000 paquets, qui comprennent 4 champs C1, C2, C3, C4 d'une longueur L de 8 bits chacun.

Cependant, l'ensemble de règles de compression RS en cours d'apprentissage a été modifié et comprend maintenant la règle R1 d'effacement de la valeur V_1^1 , règle notée $R1: [ns(V_1^1)]$. Ceci a une influence sur le calcul des potentiels de compression de champ PCC dans notre mode de réalisation, et est représentée dans la figure 2.B par la case carrée barrant le champ C1 en entier dans l'ensemble d'apprentissage E.

Le calcul des potentiels de compression de champ PCC utilise la formule $N(L - H)$, dans laquelle N est le nombre de fois que le champ apparaît dans l'ensemble d'apprentissage. Pour prendre en compte la modification de l'ensemble de règles de compression, il suffit, dans un exemple de réalisation, d'appliquer les règles de compression de l'ensemble RS à l'ensemble d'apprentissage E. Ici, l'application de la règle R1 va effacer toutes les occurrences du champ C1, puisque ce champ C1 prend une valeur unique V_1^1 effacée par la règle de compression R1. Donc, le calcul du potentiel de compression de champ $PCC(C1)$ en prenant en compte l'ajout de la règle R1 à l'ensemble de règles de compression RS va utiliser une grandeur $N=0$. On a donc $PCC(C1) = 0$ dans l'étape S1.

Il est facile de comprendre que, puisque la valeur unique prise par le champ C1 est effacée par la règle R1 pour tous les paquets de l'ensemble d'apprentissage E, ce champ n'a plus de potentiel de compression. Cette intuition est bien capturée par le calcul du potentiel de compression de champ selon le mode de réalisation présenté ici.

On voit que la formule de calcul d'un potentiel de compression de champ PCC peut s'appliquer à plusieurs contextes et en particulier ici prendre en compte l'évolution de l'ensemble de règles de compression RS au cours des étapes d'apprentissage successives.

Les autres calculs de potentiels de compression de champ $PCC(C2)$, $PCC(C3)$ et $PCC(C4)$ sont inchangés par rapport aux calculs réalisés dans la première étape S0, puisque la règle R1 ne réalise aucune opération sur les valeurs prises par les champs C2, C3 et C4.

On verra dans la description des étapes suivantes comment des calculs de potentiels de compression de champ PCC prenant en compte les règles de compression appartenant à l'ensemble RS permettent de sélectionner une règle parmi l'ensemble RS de règles de compression afin de créer des règles candidates. Comme l'ensemble RS ne comprend qu'une seule règle R1 à cette étape S1, c'est cette règle R1 qui est sélectionnée dans notre exemple.

Comme vu précédemment, une fois la règle R1 sélectionnée, il faut sélectionner un champ pour créer une ou plusieurs règles candidates. Dans notre mode de réalisation, c'est le champ qui a le plus fort potentiel de compression PCC qui est choisi, et c'est donc ici le champ C2.

Comme vu précédemment, trois règles candidates vont être créées, rajoutant à la règle R1 une opération de compression sur le champ C2 sélectionné. La première sera une opération d'effacement de valeur, cette valeur étant choisie selon les potentiels de compression de valeur. Comme le champ C2 est de longueur fixe,

les potentiels de compression de valeur donnent les mêmes classements que la fréquence d'apparition. La valeur sera celle qui apparaît le plus fréquemment et donc sera la valeur V_2^1 qui apparaît avec une fréquence de 80%. La deuxième règle candidate consistera en l'ajout d'une opération de dictionnaire. Comme il n'y a pas d'opération de dictionnaire pour le champ C2, il s'agit donc d'une création qui remplacera deux valeurs prises par le champ C2, à savoir la valeur V_2^1 , dont le potentiel de compression de valeur est le plus élevé et l'une des deux valeurs V_2^2 et V_2^3 , dont les potentiels de compression de valeur sont égaux. Mettons que la valeur V_2^2 soit retenue aléatoirement. La troisième règle candidate consiste en l'ajout d'une opération d'effacement de préfixe, et on suppose que les valeurs prises par le champ C2 sont similaires à celles prises par le champ C3, et ne présentent donc pas de préfixe commun.

- 10 L'étape suivante de l'algorithme d'apprentissage va consister en retenir une règle parmi les règles candidates créées pour l'ajouter à l'ensemble de règles de compression RS. Pour cela, la fonction de récompense utilisée dans notre mode de réalisation est la compression réalisée par les règles candidates sur les paquets de l'ensemble d'apprentissage E.

- 15 La première règle candidate réalise l'effacement de la valeur V_2^1 qui a une longueur de 8 bits, et qui apparaît dans 8000 paquets de l'ensemble d'apprentissage E. La compression réalisée par cette règle candidate est donc de 64000 bits.

- 20 La deuxième règle candidate réalise le remplacement des valeurs V_2^1 et V_2^2 par un index d'un dictionnaire, qui doit donc utiliser au moins un bit. La deuxième règle candidate remplace donc une valeur d'une longueur de 8 bits par un index d'un bit. Elle réalise donc une compression de 7 bits pour chaque paquet où elle est appliquée. En l'occurrence, cette deuxième règle est appliquée pour 9000 paquets, ceux pour lesquels la valeur du champ C2 est prise par V_2^1 ou V_2^2 . La compression réalisée est donc de $7 \times 9000 = 63000$ bits.

La première règle candidate réalise donc une meilleure compression que la deuxième.

On suppose que la troisième règle candidate n'est pas efficace car les valeurs prises par le champ C2 ne présentent pas de préfixe commun.

- 25 C'est donc la première règle candidate qui va être retenue et ajoutée à l'ensemble de règles de compression RS, donnant la règle R2. Celle-ci est notée dans la figure 2.B de la façon suivante : $R2: [ns(V_2^1) + ns(V_2^2)]$, afin de montrer que l'opération d'effacement de la valeur V_2^1 , notée $ns(V_2^1)$, est ajoutée à l'opération d'effacement de la valeur V_2^2 qui était déjà présente dans R1. Nous rappelons que la règle R2 a été créée en sélectionnant la règle R1 et en lui ajoutant une opération de compression, ici d'effacement de la valeur V_2^1 .

- 30 Dans cette étape S1, contrairement à l'étape S0, la règle R2 qui vient d'être ajoutée à l'ensemble RS de règles de compression ne s'applique pas à tous les paquets de l'ensemble d'apprentissage E auxquels s'applique la règle R1. En effet, la règle R1 s'applique à tous les paquets de l'ensemble d'apprentissage E, alors que la règle R2 ne s'applique qu'aux 8000 paquets de l'ensemble d'apprentissage E dans lesquels le champ C2 prend la valeur V_2^1 . Pour cette raison, dans le mode de réalisation présenté ici, la règle R1 n'est

pas effacée de l'ensemble RS de règles de compression qui va maintenant comprendre les deux règles R1 et R2.

Dans le mode de réalisation présenté dans cet exemple, l'algorithme d'apprentissage par renforcement va utiliser une partition de l'ensemble d'apprentissage E en attribuant des sous-ensembles de l'ensemble d'apprentissage E aux règles de compression de l'ensemble RS. Tant qu'il n'y avait qu'une règle R0 puis R1 dans l'ensemble RS, ceci n'était pas important. Ici, on va attribuer les 8000 paquets de l'ensemble d'apprentissage E pour lesquels la règle R2 s'applique à celle-ci, et créer un sous-ensemble dédié, et attribuer le résidu de 2000 paquets à la règle R1.

Cette attribution des paquets de l'ensemble d'apprentissage E à des règles de l'ensemble RS est faite dans notre exemple de réalisation au moment de l'ajout d'une nouvelle règle dans l'ensemble RS de règles de compression, au moment de décider de la suppression ou non de la règle sélectionnée de l'ensemble RS. Comme la règle créée est créée à partir de la règle sélectionnée en lui ajoutant une opération de compression, la règle créée va forcément s'appliquer à un sous-ensemble des paquets auxquels la règle sélectionnée s'applique et est attribuée. Si la règle créée s'applique à tous les paquets auxquels la règle sélectionnée s'appliquait, l'algorithme d'apprentissage va attribuer tous les paquets de la règle sélectionnée à la règle créée, et il n'est pas nécessaire de conserver la règle sélectionnée dans l'ensemble RS de règles de compression. Si la règle créée ne s'applique pas à tous les paquets de la règle sélectionnée, on attribue à la règle créée les paquets auxquels elle s'applique et on conserve la règle créée dans l'ensemble RS de règles de compression et on lui attribue le résidu de paquets auxquels la règle créée ne s'applique pas.

Cette partition va permettre d'affiner le calcul des potentiels de compression de champ comme on va le voir dans les prochaines étapes.

Nous supposons là encore qu'un critère d'arrêt n'est pas satisfait et allons présenter l'étape suivante de l'algorithme d'apprentissage par renforcement.

La figure 2.C correspond à l'état S2 de l'apprentissage, tel qu'il est obtenu après la deuxième étape, et à la troisième étape de l'apprentissage par renforcement selon un mode de réalisation.

A ce stade S2, l'ensemble RS de règles de compression comprend deux règles R1 et R2. Pour réaliser une sélection parmi ces deux règles afin de créer des règles candidates, un mode de réalisation possible utilise le calcul de potentiels de compression de champ PCC selon une partition de l'ensemble d'apprentissage E en sous-ensembles attribués aux règles R1 et R2. On a vu précédemment, lors de la description de la fin de la deuxième étape, comment procéder à cette partition.

Dans notre mode de réalisation, 8000 paquets de l'ensemble d'apprentissage E ont été attribués à la règle R2, ceux pour lesquels la règle R2 s'applique, et le résidu de 2000 paquets est attribué à la règle R1. Les potentiels de compression de champ $PCC(C1)$, $PCC(C2)$, $PCC(C3)$ et $PCC(C4)$ sont alors calculés pour les deux sous-ensembles réalisant la partition de l'ensemble d'apprentissage E, en appliquant à ces paquets la règle de compression qui leur a été attribuée.

Pour le champ C1, que ce soit dans les 2000 paquets attribués à la règle R1 ou les 8000 paquets attribués à la règle R2, l'application respective des règles R1 et R2 va effacer la valeur V_1^1 prise par le champ C1. Le champ C1 n'apparaît donc dans aucun paquet des deux sous-ensembles, et la grandeur N utilisée dans la formule de calcul du potentiel de compression de champ $PCC(C1)$ est donc nulle dans les deux cas.

- 5 Autrement dit, que ce soit pour les paquets attribués à la règle R1 ou pour ceux attribués à la règle R2, le potentiel de compression de champ PCC du champ C1 est nul. Ceci se comprend bien, puisque ces deux règles R1 et R2 effacent la valeur unique V_1^1 prise par le champ C1.

Pour le champ C2, il est nécessaire de faire un calcul différencié de potentiel de compression de champ $PCC(C2)$ pour les deux sous-ensembles attribués respectivement à la règle R1 et à la règle R2.

- 10 Pour la règle R1, les paquets de l'ensemble d'apprentissage E qui lui sont attribués sont ceux auxquels la règle R2 ne s'applique pas. Ce sont donc les 2000 paquets dans lesquels le champ C2 ne prend pas la valeur V_2^1 mais, de façon équiprobable prend l'une des deux valeurs V_2^2 ou V_2^3 . Dans ces 2000 paquets, le champ C2 apparaît bien, et la grandeur N vaut donc ici 2000. La formule pour calculer le potentiel de compression de champ $PCC(C2)$ est donc :

$$15 \quad PCC(C2) = N * (L - (-f_2^2 \log_2(f_2^2) - f_2^3 \log_2(f_2^3))) = 2000 * (8 - (-0.5(-1) - 0.5(-1))) = 2000 * 7 = 14000$$

Pour la règle R2, cependant, les 8000 paquets de l'ensemble d'apprentissage E qui lui sont attribués sont ceux auxquels la règle R2 s'applique, règle qui comprend l'opération de compression consistant en l'effacement de la valeur V_2^1 . Le champ C2 est donc complètement effacé dans le sous-ensemble de paquets attribués à la règle R2. Autrement dit, le champ C2 n'apparaît pas dans ce sous-ensemble et l'on a ici une

20 grandeur N nulle et donc un potentiel de compression de champ $PCC(C2)$ nul.

Les deux valeurs pour le potentiel de compression de champ $PCC(C2)$ sont bien présentes dans la figure 2.C, à savoir 14000 pour le sous-ensemble attribué à la règle R1 et 0 pour le sous-ensemble attribué à la règle R2.

- Pour le calcul des potentiels de compression de champ $PCC(C3)$ et $PCC(C4)$, on a vu plus haut qu'on faisait l'hypothèse que les fréquences d'apparition des différentes valeurs prises par les champs C3 et C4 sont les
- 25 mêmes dans les deux sous-ensembles attribués à la règle R1 et à la règle R2. Pour quatre valeurs équiprobables, la grandeur $H(C) = -\sum_{i \in [1,K]} f_i \log_2(f_i)$ vaut $-(4 * (0.25 * (-2)))$, soit 2. Autrement dit, pour la règle R1, qui a été attribuée à 2000 paquets, les potentiels de compression de champ $PCC(C3)$ et $PCC(C4)$ valent $2000*(8-2)$, soit 12000 ; alors que pour la règle R2, qui a été attribuée à 8000 paquets, les potentiels de compression de champ $PCC(C3)$ et $PCC(C4)$ valent $8000*(8-2)$, soit 48000. Ces deux valeurs sont bien
- 30 présentées dans la figure 2.C.

L'ensemble de ces potentiels de compression de champ sont utilisés pour déterminer la règle sélectionnée.

Un mode possible de réalisation consiste à additionner les potentiels de compression de champ PCC pour une règle donnée. Dans ce cas, la règle R1 se verrait attribuer un potentiel de compression de champ PCC

de $0 + 14000 + 12000 + 12000 = 38000$ alors que la règle R2 se verrait attribuer un potentiel de compression de champ PCC valant $0 + 0 + 48000 + 48000 = 96000$. Dans ce cas, la règle R2 serait sélectionnée.

Un autre mode possible de réalisation est de calculer le potentiel de compression de champ PCC d'une règle en prenant le maximum des potentiels de compression de champ PCC pour cette règle plutôt qu'en effectuant la somme. Ce mode de réalisation nous emmène là aussi à sélectionner la règle R2.

La création de règles candidates va donc se faire à partir de la règle R2.

Pour la règle R2, les deux champs C3 et C4 présentent le même potentiel de compression de champ $PCC(C3) = PCC(C4) = 48000$ alors que les champs C1 et C2 présentent un potentiel de compression de champ PCC nul. Dans ces cas d'égalité, la sélection entre les champs C3 et C4 pour décider quelles règles candidates vont être créées peut se faire de façon aléatoire ou en suivant un autre critère, par exemple d'ordre de survenue. Supposons que le champ C3 soit sélectionné.

Trois règles candidates sont alors créées en partant de la règle R2 et en lui ajoutant une opération de compression travaillant sur les valeurs prises par le champ C3. On rappelle que les quatre valeurs prises par le champ C3 sont équiprobables et ne présentent pas de préfixes communs, comme on peut le voir sur la figure 2.C. La règle candidate consistant en l'effacement d'un préfixe commun n'aura donc aucune efficacité de compression.

Pour la règle candidate qui consiste en l'ajout à la règle R2 d'une opération d'effacement d'une valeur prise par le champ C3, comme les valeurs sont équiprobables et que les quatre valeurs ont la même longueur, car le champ C3 est de longueur fixe, elles auront donc les mêmes potentiels de compression de valeur. N'importe quelle valeur peut donc être sélectionnée pour être effacée. Là encore, un mode de réalisation en cas d'égalité peut être de faire un tirage aléatoire ou bien de prendre la première ou la dernière valeur survenue ou toute autre méthode de choix.

De même, pour la règle candidate qui consiste en l'ajout à la règle R2 d'une opération de remplacement de valeur par un index d'un dictionnaire, les potentiels de compression de valeur ne vont pas guider le choix des valeurs à remplacer par un index de dictionnaire. Deux valeurs seront sélectionnées aléatoirement ou selon un critère d'ordre.

La compression réalisée par l'opération d'effacement se calcule pour les 8000 paquets auxquels la règle R2 s'applique. Quelle que soit la valeur effacée, celle-ci apparaît dans un quart des 8000 paquets, donc 2000 paquets. La compression réalisée par l'opération d'effacement serait donc de $2000 * 8 = 16000$ bits.

La compression réalisée par l'opération de remplacement par un index de dictionnaire se calcule aussi pour les 8000 paquets auxquels la règle R2 s'applique. Deux valeurs sont remplacées ; soit pour la moitié des paquets en question. La compression réalisée par l'opération de remplacement par un index de dictionnaire est donc de $4000 * 7 = 28000$ bits. L'index faisant 1 bit, la compression pour chaque paquet est de 7 bits, et pas de 8 bits comme pour un effacement.

La meilleure compression est celle réalisée par l'opération de remplacement par un index. Dans la figure 2.C, une telle opération d'effacement est notée $map(V_3^1, V_3^2)$, le terme « map » venant de l'anglais *mapping*. On suppose que le tirage aléatoire des deux valeurs à remplacer par un index de dictionnaire a donné les deux valeurs V_3^1 et V_3^2 . La règle R3 complète comprend donc trois opérations, à savoir les deux opérations d'effacement déjà présentes dans R2 auxquelles on ajoute la nouvelle opération, et est notée de la façon suivante : $R3: [ns(V_1^1) + ns(V_2^1) + map(V_3^1, V_3^2)]$.

La règle R3 est donc rajoutée à l'ensemble de règles de compression RS. Elle s'applique à 4000 paquets de l'ensemble d'apprentissage E (la moitié de ceux auxquels la règle R2 s'appliquait) qui vont lui être attribués pour le calcul des potentiels de compression de champ PCC à la prochaine étape. La règle R2 conserve 4000 paquets, ceux auxquels la règle de compression R3 ne s'applique pas, et n'est pas retirée de l'ensemble de règles de compression RS. Enfin, les 2000 paquets attribués à la règle R1 lui restent attribués.

La figure 2.D correspond à l'état S2' de l'apprentissage. Cet état correspond à l'état obtenu après la deuxième étape. Nous avons vu que, dans cet état, la règle R2 est sélectionnée pour créer des règles candidates, et que, pour cette règle, les champs C3 et C4 ont le même potentiel de compression de champ, à savoir $PCC(C3) = PCC(C4) = 48000$.

On a vu à l'étape précédente décrite dans la figure 2.C comment se déroulait l'algorithme d'apprentissage si les règles candidates construites à partir de la règle R2 cherchaient à compresser le champ C3. On va voir ici le déroulement de l'algorithme d'apprentissage selon un exemple de réalisation si c'est le champ C4 qui est choisi plutôt, sachant que ces deux champs ont le même potentiel de compression de champ PCC.

Dans ce cas, on rappelle que les quatre valeurs prises par le champ C4 sont équiprobables, et que la compression réalisée par les règles candidates s'estime sur les 8000 paquets attribués à la règle sélectionnée R2.

La première règle candidate sera la règle R2 à laquelle est ajoutée une opération d'effacement d'une valeur. Sachant que les quatre valeurs prises par le champ C4 sont équiprobables, les potentiels de compression de valeurs seront égaux. En choisissant n'importe laquelle des quatre valeurs équiprobables, l'ajout d'une opération d'effacement emmènerait un effacement d'une valeur de longueur 8 bits apparaissant dans 2000 paquets (un quart des 8000 paquets attribués à R2), soit une compression de 16000 bits.

La deuxième règle candidate sera la règle R2 à laquelle est ajoutée une opération de remplacement de valeur par un index de dictionnaire. Comme vu pour la figure 2.C, la compression réalisée par l'opération de remplacement par un index de dictionnaire se calcule pour les 8000 paquets auxquels la règle R2 s'applique. Deux valeurs sont remplacées ; soit pour la moitié des paquets en question. La compression réalisée par l'opération de remplacement par un index de dictionnaire est donc de $4000 * 7 = 28000$ bits. L'index faisant 1 bit, la compression pour chaque paquet est de 7 bits, et pas de 8 bits comme pour un effacement.

La troisième règle candidate sera la règle R2 à laquelle est ajoutée une opération d'effacement d'un préfixe commun. Cette règle candidate est l'occasion d'illustrer le fonctionnement des potentiels de compression de bit de champ.

- 5 Ces potentiels sont calculés pour les 8 rangs numérotés de 0 à 7, pour les valeurs prises par le champ C4 dans les paquets concernés de l'ensemble d'apprentissage. Ces quatre valeurs V_4^1 , V_4^2 , V_4^3 et V_4^4 sont équiprobables et présentent la même valeur pour les 6 premiers bits, qui valent tous 1. La fréquence d'apparition du bit 1 au rang 0 vaut donc 1, et réciproquement, la fréquence d'apparition du bit 0 au rang 0 est 0. Comme un logarithme d'une fréquence nulle n'est pas défini, l'estimateur d'entropie d'un bit de champ calculé suivant la formule $H(b_i) = -(f_i \cdot \log_2(f_i) + (1 - f_i) \cdot \log_2(1 - f_i))$ vaut par convention 0 dans ce cas, et le potentiel de compression de bit de champ pour le rang 0 vaut alors 1.

Les potentiels de compression de bit de champ valent de même 1 pour les 6 premiers rangs, puisque les bits des six premiers rangs des valeurs prises par le champ C4 sont identiques dans tous les paquets considérés. En l'occurrence, les bits valent tous 1, mais le résultat est le même, si, pour un rang donné, tous les bits valent 0.

- 15 Pour les rangs 7 et 8, les bits 0 et 1 sont équiprobables puisque les quatre valeurs V_4^1 , V_4^2 , V_4^3 et V_4^4 sont aussi équiprobables. Les potentiels de compression de bit de champ pour ces rangs 7 et 8 valent alors :

$$1 - (-(0.5 * \log_2 0.5 + (1 - 0.5) * \log_2(1 - 0.5))) = 1 - (-(0.5 * (-1) + 0.5 * (-1))) = 1 - 1 = 0$$

Le potentiel de compression de bit de champ s'annule en effet pour un rang où les bits 0 et 1 sont équiprobables.

- 20 En utilisant la méthode où les potentiels de compression de bits de champ sont multipliés entre eux, en partant du bit de rang 0, et en utilisant un seuil, on voit comment il est possible de détecter que des paquets présentent des préfixes communs pour certains champs. En l'occurrence, dans l'exemple présenté ici, le produit des potentiels de compression de bits de champ passe de 1 à 0 entre le rang 6 et le rang 7. Il est donc facile de voir que la règle candidate créée en ajoutant une opération d'effacement de préfixe à la règle R2
- 25 consistera à effacer les six premiers bits des valeurs prises par le champ C4. Cette règle R3 est notée de la façon suivante : $R3: [ns(V_1^1) + ns(V_2^1) + msb(C4,6)]$, l'acronyme « msb » voulant dire *most significant bits*, c'est-à-dire le terme anglais pour indiquer les bits de préfixe, C4 indiquant que la règle s'applique aux valeurs prises par le champ C4 et 6, que les 6 premiers bits sont concernés.

- 30 La compression réalisée par cette règle candidate est l'effacement de six bits des valeurs prises par le champ C4 dans les 8000 paquets attribués à la règle R2, puisque le préfixe commun sur 6 bits est présent pour les quatre valeurs prises par le champ C4. La compression réalisée par cette troisième règle candidate est donc de $8000 * 6 = 48000$ bits et c'est bien cette règle R3 qui est ajoutée à l'ensemble de règles de compression RS dans cette étape S2', qui diffère de l'étape S2 en ce que la création d'une règle candidate se fait en ciblant le champ C4 au lieu du champ C3.

Après l'étape S2' comme après l'étape S2, d'autres étapes de l'algorithme d'apprentissage de l'ensemble de règles de compression RS peuvent avoir lieu qui vont rajouter des opérations de compression prenant en considération les potentiels de compression restant pour finir par obtenir une compression efficace de l'ensemble d'apprentissage E.

5 En ce qui concerne l'arrêt de l'algorithme d'apprentissage, plusieurs possibilités existent.

La première possibilité consiste à calculer la compression réalisée par une nouvelle règle ajoutée à l'ensemble de règles de compression RS et à ne l'ajouter et itérer de nouveau l'algorithme d'apprentissage que si l'ajout de cette nouvelle règle améliore la compression réalisée sur les paquets de l'ensemble d'apprentissage. On pourrait croire que l'ajout d'une règle de compression à l'ensemble de règles de compression RS va forcément améliorer la compression réalisée sur l'ensemble d'apprentissage E, mais il faut prendre en compte qu'un identifiant de la règle de compression utilisée pour compresser un en-tête de paquet de données doit être transmis en même temps que le paquet compressé afin de pouvoir réaliser la décompression. Pour un ensemble RS comprenant deux règles de compression, un seul bit suffit à identifier la règle de compression utilisée. Pour quatre règles, deux bits sont nécessaires, et ainsi de suite. Nous n'avons pas pris en compte cet effet quand nous avons indiqué des calculs de compression pour sélectionner parmi les règles candidates, mais cet effet est à prendre en compte pour déterminer un critère d'arrêt de l'algorithme d'apprentissage. De plus, cet effet est rétroactif, c'est-à-dire que des calculs de compression qui sont réalisés lorsque l'ensemble RS de règles de compression comprend quatre règles, qui peuvent être désignées par deux bits, doivent être recommencés pour toutes les règles de compression quand l'ensemble RS passe à cinq règles, ce qui nécessite donc une désignation de toutes les règles par trois bits au moins.

Une autre possibilité consiste à donner une taille maximale à l'ensemble de règles de compression RS. Par exemple, il peut être décidé de limiter la taille de l'ensemble RS à 8 règles de compression, afin de s'assurer de pouvoir identifier les règles de compression utilisées pour compresser des en-têtes de paquets de données avec 3 bits seulement.

25 Ces deux possibilités peuvent être combinées, à savoir qu'un nombre maximal de règles de compression peut être défini, mais l'algorithme d'apprentissage peut s'arrêter avant que ce nombre maximal soit atteint parce que la compression réalisée par la dernière règle ajoutée est nulle ou inférieure à un seuil donné.

La figure 3, quant à elle, représente un autre exemple d'étape de calcul de l'algorithme d'apprentissage par renforcement qui est mis en œuvre dans cet exemple d'obtention d'un ensemble RS de règles de compression. Cet exemple illustre l'intérêt de calculer un potentiel de compression de champ corrigé par un facteur.

Dans l'exemple de la figure 3, nous sommes à l'étape S0 de l'algorithme d'apprentissage par renforcement. L'ensemble d'apprentissage E comprend $N = 100$ paquets. Les en-têtes de ces 100 paquets sont formés de deux champs C1 et C2 de longueur différente, contrairement à l'exemple de la figure 2 où les quatre champs

étaient de la même longueur. Le champ C1 est de longueur $L = 2$; le champ C2 est de longueur $L = 10$, les deux longueurs étant exprimées en nombre de bits. Comme dans l'exemple précédent, les sections de paquets que l'on va chercher à compresser sont les deux champs C1 et C2.

Quant aux valeurs prises par les champs dans l'ensemble d'apprentissage E, le champ C1 prend une seule valeur, donc la fréquence d'apparition de cette valeur V_1^1 vaut $f_1^1 = 100\%$. Pour le champ C2, en revanche, il présente une valeur différente dans les 100 paquets de l'ensemble d'apprentissage E. Pour le champ C2, on a donc 100 fréquences d'apprentissage qui valent toutes 1%, ce qui est noté dans la figure 3 : $f_2^1 = f_2^2 = \dots = f_2^{100} = 1\%$.

On rappelle que, dans notre exemple de réalisation, la formule pour calculer un potentiel de compression de champ PCC est : $PCC = N(L - H)$ où N est la taille de l'ensemble, L la longueur de la section et H un estimateur de l'entropie. La formule pour calculer l'estimateur d'entropie est : $H = -\sum_{i \in [1,K]} f_i \log_2(f_i)$ dans laquelle K est le nombre de valeurs différentes $V_{i \in [1,K]}$ que prend la section dans l'ensemble d'apprentissage et f_i est la fréquence d'apparition de la valeur V_i .

L'estimateur d'entropie cherche, comme son nom l'indique, à estimer l'entropie que semble manifester les valeurs prises par la section au vu de l'ensemble d'apprentissage E. Plus l'entropie est faible, plus les valeurs prises par la section seront ordonnées et donc plus il sera facile de les compresser, par exemple avec une règle d'effacement de la valeur, et donc plus la section présente de l'intérêt à être compressé, et donc plus il est intéressant que l'algorithme d'apprentissage cherche à créer des règles candidates pour cette section, la meilleure règle étant ajoutée à l'ensemble de règles de compression RS.

Par exemple, quand une section prend une seule valeur dans l'ensemble d'apprentissage E, la fréquence d'apparition de cette unique valeur vaut donc 100%, ou 1, le logarithme de cette fréquence vaut donc 0 et l'estimateur d'entropie aussi. L'estimateur d'entropie est donc à son minimum. Le potentiel de compression de champ vaut alors $PCC = N * L$, comme c'est le cas pour le champ C1 dans l'exemple de la figure 3. On a ici : $PCC(C1) = 100 * 2 = 200$, ce qui est le potentiel maximum pour un champ de longueur 2 qui apparaît dans 100 paquets. En effet, la compression du champ C1 sera faite par une règle d'effacement de la valeur unique prise par C1 dans l'ensemble d'apprentissage E, soit une compression maximale de 200 bits.

Un défaut de l'estimateur d'entropie, qui apparaît dans l'exemple de la figure 3, est que cette grandeur grandit à un rythme trop faible même en présence d'une entropie maximale. C'est ce qu'on voit dans le champ C2. L'entropie prise par les valeurs du champ C2 dans l'ensemble d'apprentissage E est maximale, puisqu'on a ici 100 valeurs différentes pour les 100 paquets, les 100 valeurs ont donc la même fréquence d'apparition, à savoir 1%, c'est-à-dire 0.01. Le logarithme de cette fréquence vaut -6.644. Le calcul du potentiel de compression de champ est détaillé sur la figure et on voit que l'estimateur d'entropie vaut finalement 6.644, ce qui est la valeur maximale qu'il peut prendre dans un ensemble d'apprentissage E de 100 paquets. Mais comme le champ C2 est de longueur 10, le potentiel de compression de champ de C2 vaut $PCC(C2) = 100 * (10 - 6.644) = 335,6$ qui est donc supérieur au potentiel de compression du champ C1, $PCC(C1)$, qui vaut

200. L'estimateur d'entropie ne grandit pas assez pour compenser l'effet de la plus grande longueur du champ C2 comparée à celle du champ C1.

On peut remarquer que si la longueur du champ C2 était $L = 8$ bits au lieu de valoir 10, alors le potentiel de compression de champ de C2 vaudrait $PCC(C2) = 100 * (8 - 6.644) = 135,6$, soit moins que le potentiel de compression de champ $PCC(C1) = 200$. Autrement dit, le score faible de l'estimateur d'entropie ne serait pas ici contrebalancé par la longueur du champ, alors que le champ C2 serait quand même de longueur 8 bits, à comparer à la longueur de 2 bits de C1.

Dans le cas où la longueur du champ C2 est de 10 bits, utiliser le potentiel de compression de champ PCC tel qu'avec sa formule initiale conduirait donc à valoriser le champ C2, ce qui serait inefficace, car la compression emmenée par une première règle serait par exemple une règle de remplacement par l'index d'un dictionnaire, qui remplacerait donc deux valeurs de 10 bits par un index d'un bit, soit une compression de 18 bits dans l'ensemble d'apprentissage E pour les deux valeurs concernées par le dictionnaire prises par le champ C2, parmi les 100 valeurs prises, à comparer aux 200 bits de compression obtenus par la règle d'effacement de la valeur unique V_1^1 prise par le champ C1.

Pour corriger cet effet, il est préférable d'introduire un facteur de correction CORR au potentiel de compression de champ prenant en compte le nombre K de valeurs différentes que prend le champ C. La formule la plus simple pour le facteur de correction CORR est $CORR = 1 - K/N$, où K est le nombre de valeurs différentes prises par le champ et N la taille de l'ensemble concerné. Plutôt que d'utiliser le potentiel de compression de champ PCC tel que donné par la formule pour sélectionner le champ à compresser, on va le multiplier par le facteur de correction CORR.

Pour le champ C1, comme ce champ prend une valeur unique dans l'ensemble d'apprentissage E, le facteur de correction CORR vaut $1 - 1/100$ soit 0,99 et le potentiel de compression corrigé $PCC(C1) * CORR$ vaut 198.

Pour le champ C2, comme ce champ prend 100 valeurs différentes dans les 100 paquets de l'ensemble d'apprentissage E, le facteur de correction CORR vaut $1 - 100/100$ soit 0 et le potentiel de compression corrigé $PCC(C2) * CORR$ est annulé aussi.

La figure 3 montre enfin l'impact du choix d'appliquer ou non le facteur de correction CORR à l'algorithme d'apprentissage. Nous sommes à l'étape initiale S0 où l'ensemble de règles de compression ne comprend que la règle R0 qui n'applique aucune opération de compression aux paquets de l'ensemble E.

Si le choix est fait de ne pas utiliser de facteur de correction CORR, ce qui est indiqué dans la figure 3 par la ligne No-corr, on verra qu'après sélection de la règle R0, c'est le champ C2 qui sera sélectionné pour créer les règles candidates. Puis, si l'on suppose qu'aucun préfixe commun significatif n'apparaît parmi les 100 valeurs prises par le champ C2 dans l'ensemble d'apprentissage E, la meilleure règle R1 parmi les candidates sera celle de création d'un dictionnaire et de remplacement de deux valeurs par leurs index.

Si le choix est fait au contraire d'utiliser le facteur de correction CORR, ce qui est indiqué dans la figure 3 par la ligne Corr, c'est bien le champ C1 qui sera sélectionné, et la règle ajoutée R1' sera celle d'effacement de la valeur unique V_1^1 prise par le champ C1. Dans ce cas, comme pour l'exemple de calcul de la figure 2, la règle R0 peut être retirée de l'ensemble d'apprentissage car tous les paquets traités par la règle R0 sont aussi traités par R1'. Ceci est indiqué par la case barrée placée sur la règle R0 dans la définition de l'ensemble de règles de compression RS.

On a vu un exemple de formule pour calculer le facteur de correction CORR mais bien d'autres exemples sont possibles. Par exemple, on peut dire de façon générale que le facteur de correction CORR vaut $\alpha(1 - \frac{K-\beta}{N+\gamma})$ dans lequel N est le nombre de fois que la section apparaît dans l'ensemble d'apprentissage ; K est le

- 10 nombre de valeurs différentes que prend la section dans l'ensemble d'apprentissage ; et α , β et γ sont des coefficients positifs choisis. L'exemple vu précédemment où CORR vaut $1 - K/N$ est obtenu alors pour les valeurs $\alpha = 1$, $\beta = 0$ et $\gamma = 0$.

D'autres choix de valeurs sont possibles pour les coefficients positifs α , β et γ qui peuvent prendre par exemple les valeurs $\alpha = 1$, $\beta = 1$ et $\gamma = 0$.

- 15 Dans ce cas, le facteur de correction prend la valeur $1 - \frac{K-1}{N}$ et vaut donc exactement 1 quand la section prend une seule valeur, c'est-à-dire pour $K = 1$. A l'opposé, le facteur de correction ne s'annule pas quand $K = N$ ce qui peut être intéressant pour conserver un certain potentiel de compression même sur une section qui semble quand même être très aléatoire puisque toutes les valeurs prises par la section dans l'ensemble d'apprentissage sont différentes.
- 20 Des formules plus complexes, non affines, peuvent être utilisées pour définir le facteur de correction CORR en prenant en compte les valeurs K et N, en faisant en sorte que le facteur de correction CORR doit être toujours positif, de préférence compris entre 0 et 1, et doit être plus faible quand le nombre K de valeurs différentes prises par la section parmi les N paquets est grand.

- 25 La figure 4, quant à elle, représente un autre exemple d'étape de calcul de l'algorithme d'apprentissage qui est mis en œuvre dans cet exemple d'obtention d'un ensemble RS de règles de compression. Cet exemple illustre l'impact de la taille des valeurs sur le potentiel de compression de chaque valeur.

Dans l'exemple présenté ici, l'ensemble d'apprentissage E comprend $N = 100$ paquets. La section qui nous intéresse est un champ C, qui se trouve être de longueur variable. Le champ C prend trois valeurs V1, V2 et

30 V3 dans l'ensemble d'apprentissage E.

La valeur V1 est de longueur $L1 = 2$ bits. Elle a une fréquence d'apparition de 80%, elle apparaît donc dans $N1 = 80$ paquets de l'ensemble d'apprentissage E.

La valeur V2 est de longueur $L2 = 100$ bits. Elle a une fréquence d'apparition de 15%, elle apparaît donc dans $N2 = 15$ paquets de l'ensemble d'apprentissage E.

La valeur V3 est de longueur $L3 = 1000$ bits. Elle a une fréquence d'apparition de 5%, elle apparaît donc dans $N3 = 5$ paquets de l'ensemble d'apprentissage E.

- 5 La formule générale pour calculer un potentiel de compression de valeur PCV selon l'invention est $PCV = N * (L - H)$ où H est un estimateur d'entropie partielle qui est le produit de la fréquence d'apparition de la valeur et de son logarithme en base 2. Comme le champ C est de longueur variable, on doit utiliser la longueur de chaque valeur V1, V2 et V3 prise par le champ C.

- 10 La figure 4 donne le calcul des trois potentiels de compression de valeur $PCV(V1)$, $PCV(V2)$ et $PCV(V3)$ pour les trois valeurs V1, V2 et V3. On voit que le potentiel le plus élevé est celui de V3. La plus grande longueur $L3 = 1000$ bits rend cette valeur plus intéressante à compresser, par exemple en l'effaçant, que les autres valeurs, même si V3 est la valeur dont la fréquence d'apparition est la plus faible.

- 15 Cet exemple simple de calcul montre que c'est bien la longueur de la valeur qui peut être effacée par une opération de compression qui est le déterminant principal du potentiel de compression de valeur, et donc que c'est bien un potentiel de compression effectif qui est utilisé dans l'algorithme d'apprentissage selon l'invention et pas de simples fréquences d'apparition.

- 20 L'exemple de calcul montre aussi l'importance que l'ensemble d'apprentissage E reflète bien le trafic effectif qui doit être compressé. Avec une fréquence très faible, une valeur très longue sera quand même sélectionnée comme ayant un bon potentiel de compression. Si le nombre de règles de compression déployée est limité, et que cette valeur très longue n'est finalement presque jamais présente, la règle de compression qui la concerne peut finir par prendre la place d'une autre règle de compression qui s'applique à une valeur moins longue mais plus fréquente, si bien que des compressions efficaces ne seront pas réalisées par suite de la mauvaise estimation des fréquences dans l'ensemble d'apprentissage E.

- 25 La figure 5, quant à elle, représente un système SYS complet de transmission de paquets de données dans un réseau NET qui utilise un ensemble RS de règles de compression obtenu par un procédé selon l'invention.

Le système SYS présente une architecture où l'on trouve plusieurs équipements réseau, à savoir une passerelle GW et deux terminaux DVC1 et DVC2. Le système SYS de transmission selon l'invention est formé de ces trois équipements GW, DV1 et DVC2.

- 30 Ces équipements réseau sont reliés par le réseau NET. Il peut s'agir par exemple d'un réseau local où la passerelle GW permet à des terminaux DVC1, DVC2 d'accéder à d'autres réseaux plus larges, tels que le réseau Internet. Les protocoles de communication entre les équipements réseaux GW, DVC1, DVC2 peuvent être de tous types du moment qu'ils se fondent sur la transmission de paquets de données P, P'. Le réseau NET peut donc être un réseau local domestique utilisant une liaison de communication WiFi, ou un réseau de

déploiement d'objets connectés de type LPWAN utilisant une liaison de communication LoRa ou tout autre liaison adaptée à des objets connectés ou bien un réseau mobile où la passerelle GW est une station de base et les terminaux DVC1, DVC2 sont des terminaux mobiles.

5 La passerelle GW comprend une entité de gestion de la compression 100 ; une entité de gestion de la décompression 200 ; ainsi qu'une entité d'obtention 300 (non représentée sur la figure) qui comprend un module 301 d'obtention d'un potentiel de compression (non représenté sur la figure), un module APP d'obtention d'un ensemble RS de règles de compression, un module de collecte COL, et un module de déploiement DEP.

10 Le terminal DVC1 comprend quant à lui une entité de gestion de la compression 100 et une entité de gestion de la décompression 200.

15 Le terminal DVC2, quant à lui, ne comprend qu'une entité de gestion de la compression 100 et pas d'entité de gestion de la décompression 200. Il peut avoir été décidé, lors de la construction du système SYS que le terminal DVC2 ne faisait quasi exclusivement qu'émettre des paquets P et n'en recevait presque pas. Pour économiser la taille mémoire nécessaire à la construction du terminal DVC2, il peut donc avoir été décidé de ne pas intégrer une entité de gestion de la décompression 200 dans le terminal DVC2. Il ne sera donc pas possible de transmettre au terminal DVC2 un paquet P compressé selon l'invention, car il ne pourrait pas le décompresser et donc le rétablir.

On voit que le système SYS de transmission comprend bien les modules suivants :

- 20 • au moins un module d'obtention APP d'un ensemble de règles de compression RS, présent dans la passerelle GW ;
- au moins un module 101 d'application d'une opération de compression à une valeur de section d'un paquet de données P, le module 101 n'étant pas représenté dans la figure 5 mais compris dans une entité de gestion de la compression 100 présente dans la passerelle GW ou les terminaux DVC1, DVC2 ;
- 25 • au moins un module 102 de transmission d'un paquet de données P, le module 102 étant de même compris dans une entité de gestion 100 ;
- au moins un module 201 de réception d'un paquet de données P, le module 201 n'étant pas représenté dans la figure 5 mais compris dans une entité de gestion de la décompression 200 présente dans la passerelle GW ou le terminal DVC1 ;
- 30 • au moins un module 202 d'application d'une opération de décompression, le module 202 étant de même compris dans une entité de gestion 200 ;
- au moins un module de collecte COL dans le réseau NET de paquets P, P' pour former un ensemble d'apprentissage E, présent dans la passerelle GW ;
- 35 • au moins un module de déploiement DEP synchronisé dans les nœuds du réseau d'un ensemble de règles de compression RS, présent dans la passerelle GW.

La passerelle GW utilise son entité de gestion de la compression 100 pour compresser puis transmettre au terminal DVC1 un paquet de données P. Le paquet P a été compressé en utilisant une règle de compression dont l'identifiant est I. Dans le mode de réalisation illustré ici, l'entité de gestion de la compression transmet de façon conjointe l'identifiant I de la règle de compression et le paquet P compressé, ce qui est illustré par la flèche étiquetée I+P entre l'entité de gestion de la compression 100 appartenant à la passerelle GW et l'entité de gestion de la décompression 200 appartenant au terminal DVC1.

Le terminal DVC2, quant à lui, adresse un paquet compressé P', qui a été compressé grâce à une règle dont l'identifiant est I'. L'entité de gestion de la compression 100 du terminal DVC2 adresse donc une concaténation I'+P' à l'entité de gestion de la décompression 200 de la passerelle GW.

- 10 Comme toutes les entités de gestion de la compression 100 et de la décompression 200 présentes dans le système SYS disposent du même ensemble de règles de compression RS, il est facile à l'équipement réseau qui le reçoit de rétablir un paquet compressé P, P' en lui appliquant les opérations de décompression symétriques des opérations de compression qui ont été appliquées aux valeurs prises par les sections du paquet de données P, P'. Les identifiants I, I' transmis conjointement avec les paquets P, P' permettent de
- 15 retrouver la règle et donc les opérations de compression qui ont été appliquées.

Dans l'architecture du système SYS présentée dans la figure 5, la passerelle GW dispose d'un module de collecte COL qui permet à la passerelle GW de rassembler les paquets de données P, P' qui transitent par la passerelle GW. Dans une architecture de type réseau local, comme celle présentée ici, la passerelle GW est idéalement placée pour voir passer l'ensemble des paquets P, P' transitant dans le réseau NET.

- 20 Les paquets P, P' collectés par le module COL vont permettre de créer un ensemble d'apprentissage E qui sera utilisé par le module d'apprentissage APP également présent dans la passerelle GW. Le module d'apprentissage APP apprend l'ensemble de règles de compression RS en utilisant des paquets P, P' réellement collectés dans le réseau NET par le module de collecte COL, ce qui est un point permettant d'améliorer fortement la pertinence de l'apprentissage et de l'ensemble de règles de compression RS. Le
- 25 module de déploiement DEP va ensuite déployer de manière synchronisée l'ensemble de règles de compression RS dans la passerelle GW et les terminaux DVC1, DVC2 du système SYS.

- Selon un mode de réalisation, les entités de gestion de la compression 100 et de la décompression 200 peuvent disposer de plusieurs ensembles de règles de compression. Ceci permet de faciliter le déploiement synchronisé de nouveaux ensembles de règles de compression RS en permettant d'utiliser de façon
- 30 concurrente plusieurs ensembles de règles de compression RS le temps que tous les équipements présents dans le réseau NET aient bien reçu les nouveaux ensembles de règles de compression RS.

Le module de collecte COL va bien sûr transmettre au module d'apprentissage APP des paquets P, P' rétablis dans leur intégrité, et non pas des paquets compressés, car seuls des paquets rétablis permettent d'apprendre l'information sur les valeurs de section les plus intéressantes à compresser.

L'ensemble d'apprentissage E ne sera pas forcément construit avec la totalité des paquets P, P' collectés par le module COL. Des stratégies d'échantillonnage peuvent être utilisées pour construire des ensembles d'apprentissage les plus représentatifs possibles du trafic dans le réseau NET, selon les moments de transmission ou bien selon les actions des terminaux DVC1, DVC2 et de la passerelle GW.

- 5 Le système de transmission SYS remplit donc bien l'objectif technique attendu, à savoir l'application de règles de compression très efficaces aux paquets P, P' en utilisant un module d'apprentissage APP pour réaliser l'apprentissage automatique de l'ensemble de règles de compression RS. Le module de collecte COL pour alimenter le module d'apprentissage APP avec un ensemble d'apprentissage E représentatif du trafic présent dans le réseau NET et le module de déploiement DEP de façon synchronisée de l'ensemble de règles de compression RS obtenu par apprentissage permettent de s'assurer que cette compression très efficace des paquets P, P' est obtenue d'une part sans intervention humaine et d'autre part va s'adapter en cas de changement de contexte dans le réseau NET, par exemple en cas de changement d'adresses IP des équipements réseau GW, DVC1, DVC2 ou en cas de rajout de nouveaux équipements dans le réseau.

- 15 Le placement des modules de collecte COL, d'apprentissage APP et de déploiement DEP dans la passerelle GW permettent de disposer d'une architecture efficace où les fonctions liées à l'apprentissage sont centralisées dans la passerelle GW. Cette architecture est efficace car la passerelle GW voit transiter l'ensemble des paquets du réseau NET et a aussi une liaison directe avec les terminaux DVC1, DVC2 présents dans le réseau NET. Cependant, d'autres architectures sont possibles.

- 20 La figure 6, quant à elle, représente un système SYS complet de transmission de paquets de données dans un réseau NET selon l'invention, mais présentant une architecture différente de celle vue dans la figure 5.

- 25 Le système SYS comprend ici trois équipements réseaux, à savoir deux routeurs RTR1 et RTR2 et un serveur distinct SRV. Le réseau NET peut être ici par exemple un cœur de réseau IP et les deux routeurs RTR1, RTR2 peuvent former une branche d'un réseau à longue distance reliant deux points. La liaison de communication peut être alors sur une fibre optique à très grand débit, et le protocole sous-jacent peut être par exemple du Gigabit Ethernet souvent utilisé en cœur de réseau, ou tout autre protocole adapté.

Les deux routeurs RTR1, RTR2 comprennent chacun une entité de gestion de la compression 100 et de la décompression 200 qui permettent de s'échanger des paquets P, P', transmis conjointement avec des identifiants I, I' de règles de compression appartenant à l'ensemble RS.

- 30 On a vu dans la présentation de la figure 5 comment la transmission des paquets P, P' fonctionnait. On fait juste ici remarquer que le système SYS de transmission comprend un serveur SRV dédié à l'apprentissage de l'ensemble de règles de compression RS. Le routeur RTR1 comprend un module de collecte COL qui va adresser des paquets P1, P2, P3 au serveur SRV. Un module de collecte COL peut être placé dans un seul routeur RTR1 puisque celui-ci voit passer, en émission ou en réception, l'ensemble des paquets transitant sur le réseau NET. Mais on peut aussi avoir plusieurs modules de collecte COL, un dans chaque routeur RTR1,
- 35

RTR2, selon la praticité de ce déploiement. Si le serveur SRV est colocalisé avec le routeur RTR1 dans un même centre de données ou une même salle réseau, il sera beaucoup plus efficace de ne déployer un module de collecte COL que dans le routeur RTR1 et pas dans le routeur RTR2 qui peut être très éloigné dans le cas d'une transmission à longue distance.

- 5 Le serveur SRV est dédié à l'apprentissage de l'ensemble de règles de compression RS. Il comprend une entité d'obtention 300 (non représentée sur la figure) qui comprend un module APP qui va réaliser l'apprentissage proprement dit selon l'invention de l'ensemble de règles de compression RS. Le module de déploiement DEP assure le déploiement synchronisé de l'ensemble de règles de compression RS dans les deux routeurs RTR1 et RTR2.
- 10 On voit que l'invention se prête à une utilisation très large, dans toutes sortes d'architectures de réseau. D'autres architectures du système de transmission SYS sont possibles pour s'adapter à d'autres architectures du réseau NET, par exemple une architecture en étoile, ou bien en anneau. De même, les protocoles qui peuvent être optimisés grâce à la compression présentée ici sont très nombreux, à la fois en cœur de réseau ou pour des réseaux locaux ou pour des réseaux mobiles.
- 15 Signalons enfin ici que, dans le présent texte, le terme « module » peut correspondre aussi bien à un composant logiciel qu'à un composant matériel ou un ensemble de composants matériels et logiciels, un composant logiciel correspondant lui-même à un ou plusieurs programmes ou sous-programmes d'ordinateur ou de manière plus générale à tout élément d'un programme apte à mettre en œuvre une fonction ou un ensemble de fonctions telles que décrites pour les modules concernés. De la même manière, un composant
- 20 matériel correspond à tout élément d'un ensemble matériel (ou hardware) apte à mettre en œuvre une fonction ou un ensemble de fonctions pour le module concerné (circuit intégré, carte à puce, carte à mémoire, etc.).

Revendications

1. Procédé d'obtention d'un ensemble (RS) de règles de compression aptes à compresser un paquet (P) de données transmis dans un réseau (NET), le paquet (P) comprenant des sections prenant des valeurs respectives exprimées sous la forme de suite de bits, le paquet (P) étant apte à être compressé par application de règles de compression aptes à supprimer des bits d'une valeur prise par une section du paquet (P), le procédé comprenant, pour plusieurs sections données de paquets, une obtention d'un potentiel de compression (PC) pour une section donnée, relativement aux valeurs prises par la section donnée dans les paquets appartenant à un ensemble (E) de paquets de données, ensemble dit d'apprentissage, suivie par l'obtention de l'ensemble (RS) de règles de compression en fonction des potentiels de compression (PC) obtenus.
2. Procédé d'obtention d'un ensemble (RS) de règles de compression selon la revendication 1, caractérisé en ce qu'il comprend les étapes suivantes exécutées lors de l'obtention de l'ensemble (RS) de règles de compression :
 - une étape de sélection d'une règle appartenant à l'ensemble (RS) de règles de compression en utilisant au moins un potentiel de compression (PC) parmi ceux obtenus ;
 - une étape de création d'une nouvelle règle à partir de la règle sélectionnée en utilisant au moins un potentiel de compression (PC) parmi ceux obtenus ;
 - l'ajout de la nouvelle règle à l'ensemble (RS) de règles de compression ;

et en ce que lesdites étapes sont itérées, jusqu'à ce qu'un critère d'arrêt soit satisfait, en utilisant à l'itération suivante l'ensemble (RS) de règles modifié par l'ajout de la nouvelle règle.

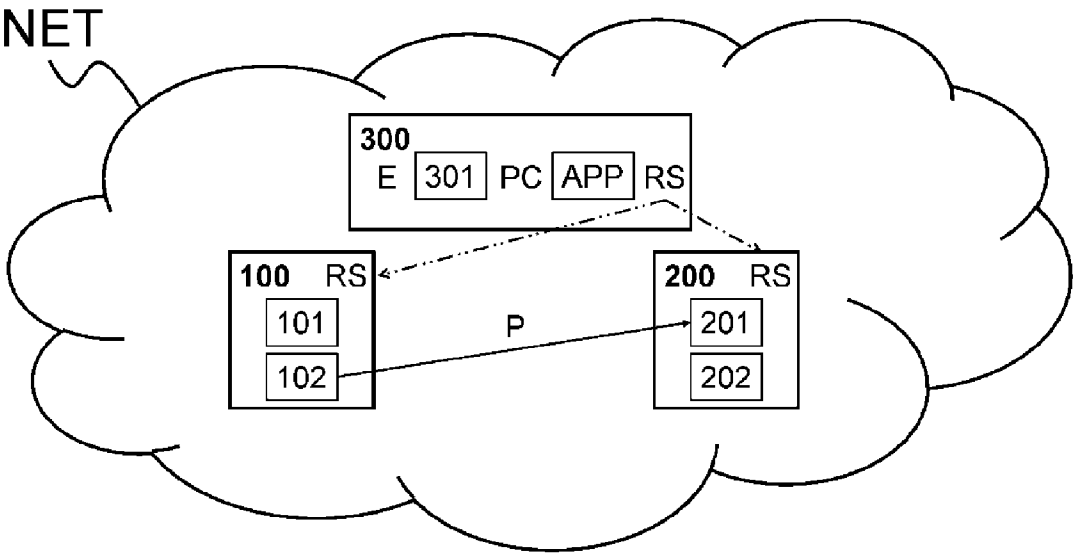
3. Procédé d'obtention d'un ensemble (RS) de règles de compression selon l'une des revendications 1 ou 2, caractérisé en ce que l'un des potentiels de compression obtenu est dit potentiel de compression de champ (PCC) et est calculé pour une section donnée de longueur fixe suivant la formule $N(L - H)$ dans laquelle N est le nombre de fois que la section apparaît dans l'ensemble dit d'apprentissage ; L est la longueur de la section exprimée en bits et H est une grandeur dite estimateur d'entropie de champ qui est calculée suivant la formule $H = -\sum_{i \in [1, K]} f_i \log_2(f_i)$ dans laquelle K est le nombre de valeurs différentes $V_{i \in [1, K]}$ que prend la section dans l'ensemble dit d'apprentissage (E) et f_i est la fréquence d'apparition de la valeur V_i .
4. Procédé d'obtention d'un ensemble (RS) de règles de compression selon l'une des revendications 1 à 3, caractérisé en ce que l'un des potentiels de compression obtenu est dit potentiel de compression de champ (PCC) et est calculé pour une section donnée de longueur variable suivant la formule $\sum_{i=1}^K N_i \cdot (L_i - H_i)$ dans laquelle K est le nombre de valeurs différentes $V_{i \in [1, K]}$ que prend la section dans l'ensemble dit d'apprentissage (E) ; N_i est le nombre de fois que la section prend la valeur V_i dans l'ensemble dit d'apprentissage (E) ; L_i est la longueur de la valeur V_i exprimée en bits et H_i est une grandeur dite estimateur d'entropie partielle d'une valeur, qui est calculée pour la valeur V_i suivant la formule $H_i = -f_i \log_2(f_i)$ dans laquelle f_i est la fréquence d'apparition de la valeur V_i dans l'ensemble dit d'apprentissage (E).

5. Procédé d'obtention d'un ensemble (RS) de règles de compression selon l'une des revendications 1 à 4, caractérisé en ce que l'un des potentiels de compression obtenu est dit potentiel de compression corrigé et est calculé pour une section donnée en multipliant le potentiel de compression de champ (PCC) par un facteur de correction $\alpha(1 - \frac{K-\beta}{N+\gamma})$ dans lequel N est le nombre de fois que la section apparaît dans l'ensemble dit d'apprentissage (E) ; K est le nombre de valeurs différentes que prend la section dans l'ensemble dit d'apprentissage (E) ; et α , β et γ sont des coefficients positifs choisis.
6. Procédé d'obtention d'un ensemble (RS) de règles de compression selon l'une des revendications 1 à 5, caractérisé en ce que l'un des potentiels de compression obtenu est dit potentiel de compression de valeur (PCV) et est calculé pour une valeur V_i d'une section donnée suivant la formule $N(L - H_i)$ dans laquelle N est le nombre de fois que la valeur V_i apparaît dans l'ensemble dit d'apprentissage (E) ; L est la longueur de la valeur V_i en bits ; et H_i est une grandeur dite estimateur d'entropie partielle d'une valeur qui est calculée suivant la formule $H_i = -f_i \log_2(f_i)$ dans laquelle f_i est la fréquence d'apparition de la valeur V_i .
7. Procédé d'obtention d'un ensemble (RS) de règles de compression selon l'une des revendications 1 à 6, caractérisé en ce que l'une des grandeurs calculée est dite potentiel de compression de bit de champ (PCB) et est calculée pour un rang i d'une section donnée, les rangs i variant de 0 à $L - 1$, L étant la longueur minimale des valeurs prises par la section dans l'ensemble dit d'apprentissage (E), ladite longueur L étant exprimée en nombre de bits, et ladite grandeur (PCB) est calculée suivant la formule $1 - H(b_i)$, dans laquelle $H(b_i)$ est une grandeur dite estimateur d'entropie d'un bit de champ qui est calculée suivant la formule $H(b_i) = -(f_i \cdot \log_2(f_i) + (1 - f_i) \cdot \log_2(1 - f_i))$, ladite formule valant par convention 0 quand f_i vaut 0 ou 1, formule dans laquelle les valeurs prises par la section dans l'ensemble dit d'apprentissage (E) sont exprimées en séquences de bits ordonnées et dans laquelle f_i est la fréquence d'apparition de l'une des valeurs 0 ou 1 pour le bit de rang i parmi les valeurs prises par la section dans l'ensemble dit d'apprentissage (E).
8. Procédé d'obtention d'un ensemble (RS) de règles de compression selon l'une des revendications 2 à 7, caractérisé en ce que l'étape de sélection d'une règle appartenant à l'ensemble (RS) de règles de compression comprend la partition de l'ensemble dit d'apprentissage (E) en sous-ensembles attribués à une règle de l'ensemble (RS) de règles de compression ; le calcul d'un potentiel de compression (PC) pour les règles de l'ensemble (RS) de règles de compression et la sélection de la règle selon son potentiel de compression (PC), le potentiel de compression (PC) d'une règle étant calculé à partir de potentiels de compressions de champ (PCC) calculés sur le sous-ensemble de l'ensemble dit d'apprentissage (E) attribué à la règle de compression.
9. Procédé d'obtention d'un ensemble (RS) de règles de compression selon l'une des revendications 2 à 8, caractérisé en ce que l'étape de création d'une nouvelle règle à partir de la règle sélectionnée s'effectue en créant une ou plusieurs règles candidates et en retenant une règle parmi les règles candidates créées selon l'efficacité de la compression réalisée par cette règle sur des paquets de l'ensemble dit d'apprentissage (E).
10. Procédé d'obtention d'un ensemble (RS) de règles de compression selon l'une des revendications 2 à 9, caractérisé en ce que l'étape de création d'une nouvelle règle à partir de la règle sélectionnée

s'effectue en choisissant une section selon son potentiel de compression de champ (PCC) et en créant des règles candidates formées en ajoutant à la règle sélectionnée une opération de compression s'appliquant aux valeurs prises par la section.

11. Entité de gestion (300) de l'obtention d'un ensemble (RS) de règles de compression aptes à compresser un paquet (P) de données transmis dans un réseau (NET), le paquet (P) comprenant des sections prenant des valeurs respectives exprimées sous la forme de suite de bits, le paquet (P) étant apte à être compressé par application de règles de compression aptes à supprimer des bits d'une valeur prise par une section du paquet (P), caractérisée en ce que l'entité de gestion (300) comprend les modules suivants :
 - un module (301) d'obtention d'un potentiel de compression (PC) pour une section donnée, relativement aux valeurs prises par la section donnée dans les paquets appartenant à un ensemble (E) de paquets de données, dit ensemble d'apprentissage ;
 - un module (APP) d'obtention de l'ensemble (RS) de règles de compression en fonction des potentiels de compression (PC) obtenus.
12. Entité de gestion (300) selon la revendication 11 comprenant de surcroît les modules suivants :
 - un module (COL) de collecte dans le réseau (NET) de paquets de données transmis pour former un ensemble d'apprentissage (E) utilisé pour obtenir l'ensemble (RS) de règles de compression ;
 - un module (DEP) de déploiement synchronisé dans les nœuds du réseau (NET) de l'ensemble (RS) de règles de compression.
13. Equipement réseau (GW, DVC1, DVC2, RTR1, RTR2) comprenant une entité de gestion (300) de l'obtention d'un ensemble (RS) de règles de compression selon la revendication 11.
14. Programme d'ordinateur apte à être mis en œuvre par une entité de gestion (300), le programme comprenant des instructions de code qui, lorsqu'il est exécuté par un processeur, réalise les étapes du procédé d'obtention défini dans la revendication 1.
15. Support de données sur lequel est enregistré un programme d'ordinateur comprenant une séquence d'instructions de code pour la mise en œuvre du procédé d'obtention conforme à la revendication 1 lorsqu'il est chargé dans et exécuté par un processeur.

[Fig 1]



[Fig 2.A]

S0:

E = $\left\{ \begin{array}{ll} N = 10000 & L = 8 \\ C1: f_1^1 = 100\% & PCC(C1) = 80000 \\ C2: f_2^1 = 80\%; f_2^2 = 10\%; f_2^3 = 10\% & PCC(C2) = 70781 \\ C3: f_3^1 = f_3^2 = f_3^3 = f_3^4 = 25\% & PCC(C3) = 60000 \\ V_3^1 = 00111111; V_3^2 = 01111111 & \\ V_3^3 = 10111111; V_3^4 = 11111111 & \\ C4: f_4^1 = f_4^2 = f_4^3 = f_4^4 = 25\% & PCC(C4) = 60000 \\ V_4^1 = 11111100; V_4^2 = 11111101 & \\ V_4^3 = 11111110; V_4^4 = 11111111 & \end{array} \right\}$

RS = {R0}

$R0 \Rightarrow R1: [ns(V_1^1)]$ **RS = {~~R0~~; R1: [ns(V₁¹)]}**

[Fig 2.B]

$$\begin{array}{l}
 \mathbf{S1:} \\
 \mathbf{E =} \left\{ \begin{array}{ll}
 \begin{array}{l}
 \text{N = 10000} \\
 \boxed{\text{C1: } f_1^1 = 100\%} \\
 \text{C2: } f_2^1 = 80\%; f_2^2 = 10\%; f_2^3 = 10\% \\
 \text{C3: } f_3^1 = f_3^2 = f_3^3 = f_3^4 = 25\% \\
 V_3^1 = 00111111; V_3^2 = 01111111 \\
 V_3^3 = 10111111; V_3^4 = 11111111 \\
 \text{C4: } f_4^1 = f_4^2 = f_4^3 = f_4^4 = 25\% \\
 V_4^1 = 11111100; V_4^2 = 11111101 \\
 V_4^3 = 11111110; V_4^4 = 11111111
 \end{array}
 &
 \begin{array}{l}
 \text{L = 8} \\
 \text{PCC(C1) = 0} \\
 \text{PCC(C2) = 70781} \\
 \text{PCC(C3) = 60000} \\
 \text{PCC(C4) = 60000}
 \end{array}
 \end{array} \right\}
 \end{array}$$

$$\mathbf{RS = \{R1: [ns(V_1^1)]\}}$$

$$R1 \Rightarrow R2: [ns(V_1^1) + ns(V_2^1)]$$

$$\mathbf{RS = \{R1: [ns(V_1^1)]; R2: [ns(V_1^1) + ns(V_2^1)]\}}$$

[Fig 2.C]

$$\begin{array}{l}
 \mathbf{S2:} \\
 \mathbf{E =} \left\{ \begin{array}{ll}
 \begin{array}{l}
 \text{N = 10000} \\
 \boxed{\text{C1: } f_1^1 = 100\%} \\
 \boxed{\text{C2: } f_2^1 = 80\%}; f_2^2 = 10\%; f_2^3 = 10\% \\
 \text{C3: } f_3^1 = f_3^2 = f_3^3 = f_3^4 = 25\% \\
 V_3^1 = 00111111; V_3^2 = 01111111 \\
 V_3^3 = 10111111; V_3^4 = 11111111 \\
 \text{C4: } f_4^1 = f_4^2 = f_4^3 = f_4^4 = 25\% \\
 V_4^1 = 11111100; V_4^2 = 11111101 \\
 V_4^3 = 11111110; V_4^4 = 11111111
 \end{array}
 &
 \begin{array}{l}
 \text{L = 8} \\
 \text{R1-R2:PCC(C1) = 0} \\
 \text{R1:PCC(C2) = 14000} \\
 \text{R2:PCC(C2) = 0} \\
 \text{R1:PCC(C3) = 12000} \\
 \text{R2:PCC(C3) = 48000} \\
 \text{R1:PCC(C4) = 12000} \\
 \text{R2:PCC(C4) = 48000}
 \end{array}
 \end{array} \right\}
 \end{array}$$

$$\mathbf{RS = \{R1: [ns(V_1^1)]; R2: [ns(V_1^1) + ns(V_2^1)]\}}$$

$$R2 \Rightarrow R3: [ns(V_1^1) + ns(V_2^1) + \text{map}(V_3^1, V_3^2)]$$

$$\mathbf{RS = \{R1: [...]; R2: [...]; R3: [ns(V_1^1) + ns(V_2^1) + \text{map}(V_3^1, V_3^2)]\}}$$

[Fig 2.D]

$$\begin{array}{l}
 \mathbf{S2'}: \\
 \mathbf{E} = \left\{ \begin{array}{ll}
 \begin{array}{l}
 N = 10000 \\
 \boxed{C1: f_1^1 = 100\%} \\
 \boxed{C2: f_2^1 = 80\%}; f_2^2 = 10\%; f_2^3 = 10\% \\
 C3: f_3^1 = f_3^2 = f_3^3 = f_3^4 = 25\% \\
 V_3^1 = 00111111; V_3^2 = 01111111 \\
 V_3^3 = 10111111; V_3^4 = 11111111 \\
 C4: f_4^1 = f_4^2 = f_4^3 = f_4^4 = 25\% \\
 V_4^1 = 11111100; V_4^2 = 11111101 \\
 V_4^3 = 11111110; V_4^4 = 11111111
 \end{array}
 &
 \begin{array}{l}
 L = 8 \\
 R1-R2:PCC(C1) = 0 \\
 R1:PCC(C2) = 14000 \\
 R2:PCC(C2) = 0 \\
 R1:PCC(C3) = 12000 \\
 R2:PCC(C3) = 48000 \\
 R1:PCC(C4) = 12000 \\
 R2:PCC(C4) = 48000
 \end{array}
 \end{array} \right.
 \end{array}$$

$$\begin{aligned}
 \mathbf{RS} &= \{R1: [ns(V_1^1)]; R2: [ns(V_1^1) + ns(V_2^1)]\} \\
 R2 &\Rightarrow R3: [ns(V_1^1) + ns(V_2^1) + msb(C4, 6)] \\
 \mathbf{RS} &= \{R1: [...]; R2: [...]; R3: [ns(V_1^1) + ns(V_2^1) + msb(C4, 6)]\}
 \end{aligned}$$

[Fig 3]

$$\begin{array}{l}
 \mathbf{S0}: \\
 \mathbf{E} = \left\{ \begin{array}{l}
 N = 100 \\
 C1 : L = 2 : f_1^1 = 100\% \\
 PCC(C1) = 100 * (2 - 0) = 200 \\
 CORR = 1 - 1/100 = 0,99 \\
 PCC(C1) * CORR = 198 \\
 C2 : L = 10 : f_2^1 = f_2^2 = \dots = f_2^{100} = 1\% \\
 PCC(C2) = 100 * (10 - (-100 * 0,01 (\log(0,01)))) \\
 PCC(C2) = 100 * (10 - 100 * 0,06644) = 100 * (10 - 6,644) \\
 PCC(C2) = 335,6 \\
 CORR = 1 - 100/100 = 0 \\
 PCC(C2) * CORR = 0
 \end{array} \right.
 \end{array}$$

$$\mathbf{RS} = \{R0\}$$

$$\begin{array}{ll}
 \text{No-corr} : R0 \Rightarrow R1: [map(V_2^1, V_2^2)] & \mathbf{RS} = \{R0 ; R1: [map(V_2^1, V_2^2)]\} \\
 \text{Corr} : R0 \Rightarrow R1': [ns(V_1^1)] & \mathbf{RS} = \{\boxed{R0}\}; R1': [ns(V_1^1)]
 \end{array}$$

[Fig 4]

S0:

E =

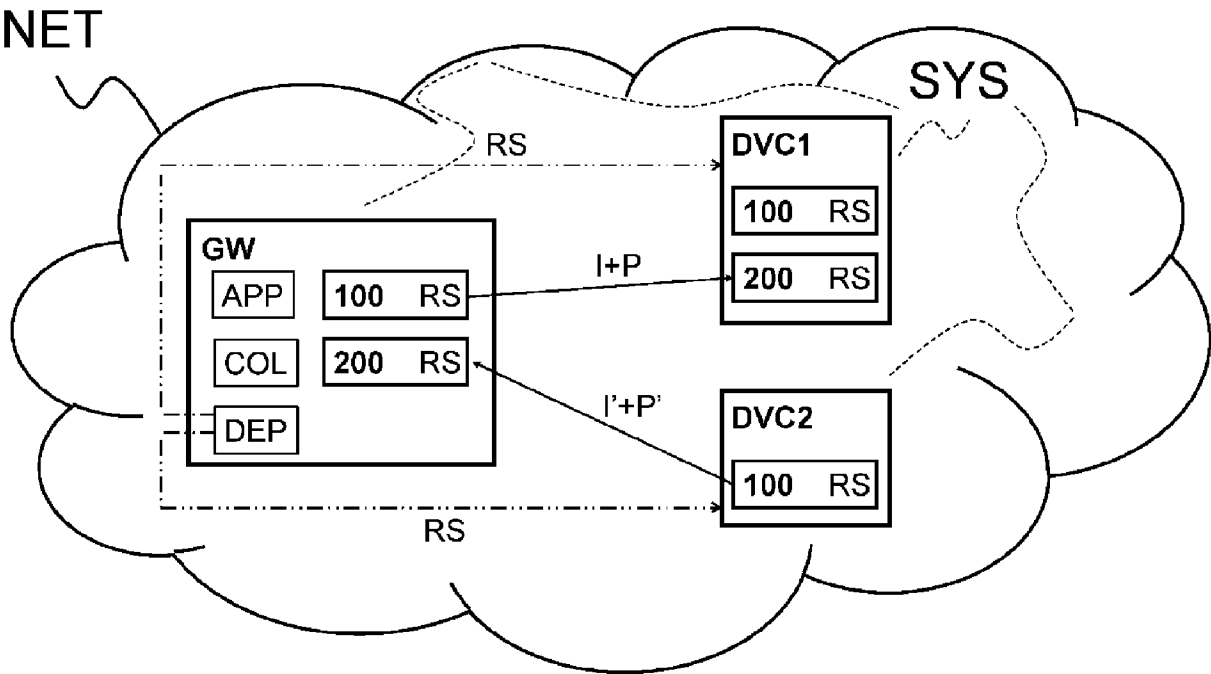
N = 100

C : V1 : L1 = 2 : $f^1 = 80\%$
PCV(V1) = $N1 * (L1 - H1)$
PCV(V1) = $80 * (2 - (-0,8 * \log(0,8)))$
PCV(V1) = $80 * (2 - (-0,8 * (-0,3219))) = 139,4$

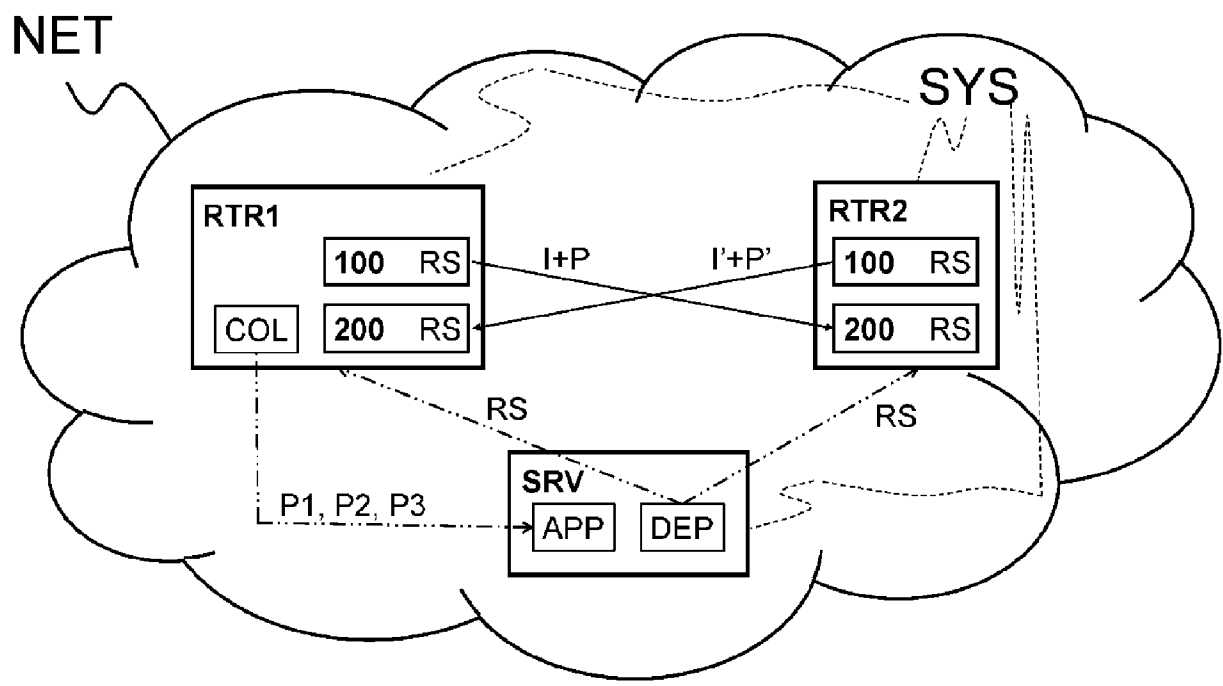
C : V2: L2 = 100 : $f^2 = 15\%$
PCV(V2) = $N2 * (L2 - H2)$
PCV(V2) = $15 * (100 - (-0,15 * \log(0,15)))$
PCV(V2) = $15 * (100 - (-0,15 * (-2,737))) = 1494$

C : V3 : L3 = 1000 : $f^3 = 5\%$
PCC(V3) = $N3 * (L3 - H3)$
PCV(V3) = $5 * (1000 - (-0,05 * \log(0,05)))$
PCV(V3) = $5 * (1000 - (-0,05 * (-4,322))) = 4999$

[Fig 5]



[Fig 6]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/EP2024/065211

A. CLASSIFICATION OF SUBJECT MATTER <i>H04L 69/04</i> (2022.01)i; <i>G06N 20/00</i> (2019.01)i; <i>H03M 7/30</i> (2006.01)i; <i>G06N 3/006</i> (2023.01)i According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) H04L; G06N; H03M Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) EPO-Internal		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X A	US 2021110292 A1 (MINABURO ANA [FR] ET AL) 15 April 2021 (2021-04-15) paragraphs [0018] - [0131], [0147]; claims 1,4; figures	1,2,9-15 3-8
A	TOMOSKOZI MATE ET AL. "Packet Header Compression: A Principle-Based Survey of Standards and Recent Research Studies" <i>IEEE COMMUNICATIONS SURVEYS & TUTORIALS, IEEE</i> , Vol. 24, No. 1, 20 January 2022 (2022-01-20), pages 698-740 DOI: 10.1109/COMST.2022.3144473 XP011901225 sections I.A, II.B.2)-6), II.C, II.F, III.H	1-15
☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.		
* Special categories of cited documents: “A” document defining the general state of the art which is not considered to be of particular relevance “E” earlier application or patent but published on or after the international filing date “L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) “O” document referring to an oral disclosure, use, exhibition or other means “P” document published prior to the international filing date but later than the priority date claimed “T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention “X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone “Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art “&” document member of the same patent family		
Date of the actual completion of the international search 30 August 2024		Date of mailing of the international search report 13 September 2024
Name and mailing address of the ISA/EP European Patent Office p.b. 5818, Patentlaan 2, 2280 HV Rijswijk Netherlands (Kingdom of the) Telephone No. (+31-70)340-2040 Facsimile No. (+31-70)340-3016		Authorized officer Scappazzoni, E Telephone No.

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/EP2024/065211

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
US	2021110292	A1	15 April 2021	CN	110651457	A	03 January 2020
				EP	3603000	A1	05 February 2020
				ES	2864161	T3	13 October 2021
				FR	3064865	A1	05 October 2018
				US	2021110292	A1	15 April 2021
				WO	2018178242	A1	04 October 2018

RAPPORT DE RECHERCHE INTERNATIONALE

Demande internationale n°
PCT/EP2024/065211

A. CLASSEMENT DE L'OBJET DE LA DEMANDE		
INV.	H04L69/04	G06N20/00
		H03M7/30
		G06N3/006
ADD.		
Selon la classification internationale des brevets (CIB) ou à la fois selon la classification nationale et la CIB		
B. DOMAINES SUR LESQUELS LA RECHERCHE A PORTE		
Documentation minimale consultée (système de classification suivi des symboles de classement)		
H04L G06N H03M		
Documentation consultée autre que la documentation minimale dans la mesure où ces documents relèvent des domaines sur lesquels a porté la recherche		
Base de données électronique consultée au cours de la recherche internationale (nom de la base de données, et si cela est réalisable, termes de recherche utilisés)		
EPO- Internal		
C. DOCUMENTS CONSIDERES COMME PERTINENTS		
Catégorie*	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
X	US 2021/110292 A1 (MINABURO ANA [FR] ET AL) 15 avril 2021 (2021-04-15)	1, 2, 9 - 15
A	alinéas [0018] - [0131], [0147]; revendications 1, 4; figures	3 - 8

A	TOMOSKOZI MATE ET AL: "Packet Header Compression: A Principle-Based Survey of Standards and Recent Research Studies", IEEE COMMUNICATIONS SURVEYS & TUTORIALS, IEEE, vol. 24, no. 1, 20 janvier 2022 (2022-01-20), pages 698-740, XP011901225, DOI: 10.1109/COMST.2022.3144473 [extrait le 2022-02-23] Sections I.A, II.B.2) - 6), II.C, II.F, III.H	1 - 15

☐ Voir la suite du cadre C pour la fin de la liste des documents ☒ Les documents de familles de brevets sont indiqués en annexe		
* Catégories spéciales de documents cités: "A" document définissant l'état général de la technique, non considéré comme particulièrement pertinent "E" document antérieur, mais publié à la date de dépôt international ou après cette date "L" document pouvant jeter un doute sur une revendication de priorité ou cité pour déterminer la date de publication d'une autre citation ou pour une raison spéciale (telle qu'indiquée) "O" document se référant à une divulgation orale, à un usage, à une exposition ou tous autres moyens "P" document publié avant la date de dépôt international, mais postérieurement à la date de priorité revendiquée "T" document ultérieur publié après la date de dépôt international ou la date de priorité et n'appartenant pas à l'état de la technique pertinent, mais cité pour comprendre le principe ou la théorie constituant la base de l'invention "X" document particulièrement pertinent:: l'invention revendiquée ne peut être considérée comme nouvelle ou comme impliquant une activité inventive par rapport au document considéré isolément "Y" document particulièrement pertinent:: l'invention revendiquée ne peut être considérée comme impliquant une activité inventive lorsque le document est associé à un ou plusieurs autres documents de même nature, cette combinaison étant évidente pour une personne du métier "&" document qui fait partie de la même famille de brevets		
Date à laquelle la recherche internationale a été effectivement achevée		Date d'expédition du présent rapport de recherche internationale
30 août 2024		13/09/2024
Nom et adresse postale de l'administration chargée de la recherche internationale		Fonctionnaire autorisé
Office Européen des Brevets, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016		Scappazzoni, E

Renseignements relatifs aux membres de familles de brevets

PCT/EP2024/065211

Formulaire PCT/ISA/210 (annexe familles de brevets) (avril 2005)