

(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
8 juillet 2010 (08.07.2010)

PCT

(10) Numéro de publication internationale
WO 2010/076536 A2

(51) Classification internationale des brevets :
H04L 29/08 (2006.01) **H04L 12/56** (2006.01)
H04L 29/12 (2006.01)

Issy Les Moulineaux (FR). **DINAND, Xavier** [FR/FR]; 5,
rue Pierre Perrier, F-49340 Vezins (FR).

(21) Numéro de la demande internationale :
PCT/FR2009/052713

(74) Mandataire : **RENARD, Béatrice**; France Telecom
R&D/PIV/Brevets, 38-40, rue du Général Leclerc,
F-92794 Issy Moulineaux Cedex 9 (FR).

(22) Date de dépôt international :
29 décembre 2009 (29.12.2009)

(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, BR, BW, BY, BZ,
CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO,
DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,
HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP,
KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD,
ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI,
NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD,
SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR,
TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(25) Langue de dépôt : français

(26) Langue de publication : français

(30) Données relatives à la priorité :
0859119 30 décembre 2008 (30.12.2008) FR

(71) Déposant (pour tous les États désignés sauf US) :
FRANCE TELECOM [FR/FR]; 6, place d'Alleray,
F-75015 Paris (FR).

(72) Inventeurs; et

(84) États désignés (sauf indication contraire, pour tout titre
de protection régionale disponible) : ARIPO (BW, GH,
GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM,

(75) Inventeurs/Déposants (pour US seulement) :
MIGAULT, Daniel [FR/FR]; 52, rue Hoche, F-92130

[Suite sur la page suivante]

(54) Title : METHOD FOR PROCESSING REQUESTS ISSUED BY A CLIENT

(54) Titre : PROCÈDE DE TRAITEMENT DE REQUÊTES ÉMISES PAR UN CLIENT

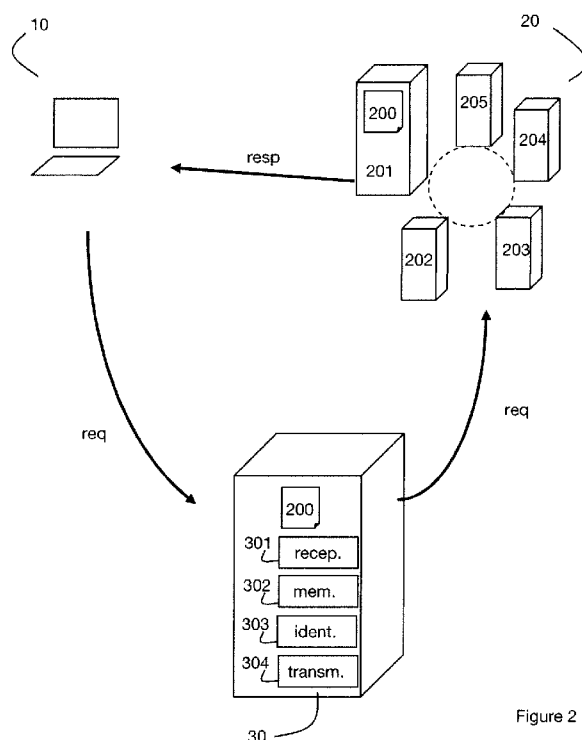


Figure 2

(57) Abstract : The invention relates to a system for processing a request to obtain data issued by a client, including: - a plurality of nodes (201, 202, 203, 204, 205) organised into a P2P network, suitable for storing a data set distributed across the various nodes and storing an identification table (200) suitable for identifying the node among the plurality of nodes that is capable of supplying the data in response to the request, characterised in that said system includes: - a proxy server (30) storing said table, to which the request is intended to be directed, said server only being suitable for, upon receiving the request: - identifying, based on the table, the node capable of supplying the data in response to the request, and - retransmitting the request to the identified node.

(57) Abrégé : L'invention concerne un système de traitement d'une requête émise par un client pour obtenir une donnée, comprenant : une pluralité de nœuds (201, 202, 203, 204, 205) organisés en réseau P2P, adaptés pour stocker un ensemble de données réparties dans

[Suite sur la page suivante]

WO 2010/076536 A2



ZW), eurasien (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), européen (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Publiée :

— *sans rapport de recherche internationale, sera republiée dès réception de ce rapport (règle 48.2.g)*

les différents nœuds et mémorisant une table d'identification (200) adaptée pour identifier, celui parmi la pluralité de nœuds qui est apte à fournir la donnée en réponse à la requête, caractérisé en ce qu'il comprend : un serveur mandataire (30) mémorisant ladite table et auquel la requête est destinée à être adressée, ledit serveur étant adapté pour, suite à la réception de la requête, uniquement : identifier, à partir de la table, le nœud apte à fournir la donnée en réponse à la requête; et retransmettre la requête au nœud identifié.

Procédé de traitement de requêtes émises par un client

L'invention concerne une technique pour traiter une requête émise par un client pour obtenir une donnée.

5

L'invention trouve une application particulièrement intéressante dans des systèmes "DNS" (de l'anglais "Domain Name System") et leur version sécurisée "DNSSEC" (pour "Domain Name System Security Extensions").

Le système DNS est un élément fondamental de l'internet qui permet
10 d'associer par exemple un nom de machine compréhensible et facilement mémorisable à une adresse IP, plus difficile à mémoriser pour un être humain. Le système DNS fonctionne comme une importante base de données.

Un succès du système DNS réside dans une répartition des données, ou enregistrements de donnée DNS, au sein de serveurs caches. Ces serveurs
15 caches gardent en mémoire des informations qui ont déjà été demandées par un ou des clients, et qui ont déjà transité par leur biais.

Le principe de fonctionnement des serveurs DNS caches, comme celui des serveurs DNSSEC caches est le suivant : un client qui souhaite résoudre un nom de domaine par l'intermédiaire d'une requête DNS, par exemple pour
20 obtenir une adresse IP associée à un nom de domaine www.monserveur.com, s'adresse en premier lieu à un serveur cache. Ce serveur cache vérifie dans un premier temps si la réponse à la requête est déjà présente dans sa mémoire cache. Dans le cas où le serveur cache détient la réponse, il envoie la réponse au client. Dans le cas où il ne détient pas la réponse en cache, il effectue une
25 résolution DNS en interrogeant un, ou des serveurs DNS de l'Internet, appelés serveurs autoritaires. Une fois la résolution effectuée, le serveur cache détient la réponse et répond au client. L'utilisation de serveurs caches permet d'économiser des transferts d'informations entre des clients et des serveurs distants, et de diminuer des temps de réponse à des requêtes DNS.

30

Cependant, la gestion des serveurs caches dans une architecture DNS n'est pas optimisée. En effet, les serveurs caches fonctionnent de manière

indépendante les uns des autres. Ainsi, plusieurs serveurs DNS caches sont susceptibles de stocker les mêmes enregistrements DNS. D'autre part, la version sécurisée du système DNS, DNSSEC, nécessite des informations supplémentaires pour gérer un enregistrement de données. Un fichier DNSSEC est donc, par nature, de taille beaucoup plus importante qu'un fichier DNS. En outre, depuis plusieurs années, on observe un accroissement régulier du nombre de requêtes DNS. Cette tendance devrait s'accroître à l'avenir avec un développement de services basés sur Internet, tels que des services de voix sur IP. Il en résulte un besoin croissant de stockage de données dans des serveurs caches, données qui sont de taille plus importante dans la version sécurisée DNSSEC. Or, si des mêmes données sont stockées dans une pluralité de serveurs caches, on comprend que l'espace mémoire des serveurs caches n'est pas gérée de manière optimale. On s'expose ainsi à des problèmes de performances, notamment à des temps de réponse à des requêtes de plus en plus importants. En définitive, cela peut conduire à une dégradation du service DNS.

En vue de faire face aux problèmes de performances tels que cités précédemment, des travaux ont étudié un remplacement d'une architecture DNS classique, par une architecture de type "P2P" (pour "Peer to Peer"). Dans une telle architecture, les données sont réparties entre les serveurs caches DNS qui forment des nœuds du réseau P2P. Dans cette architecture, les serveurs caches ne sont plus indépendants et communiquent entre eux. A cette fin, il est nécessaire de disposer d'une règle de répartition des enregistrements au sein des serveurs caches. Une mise en œuvre connue d'une telle architecture repose, pour les aspects stockage et accès aux données, sur l'utilisation d'une technologie appelée table de hachage distribuée, ou "table DHT" (DHT pour "Distributed Hash Table"), ou encore table de condensats. Une telle table permet l'identification et l'obtention d'une information dans un système réparti, comme par exemple dans certains réseaux P2P. Un exemple de protocole qui utilise une table de hachage distribuée est le protocole Pastry.

Un exemple d'architecture de l'art antérieur illustrant ces travaux est présenté en relation avec la figure 1. Dans cette architecture, chaque serveur

cache 201, 202, 203, 204, 205 est un nœud DHT du réseau P2P 20, responsable d'un ensemble de données. Un premier nœud DHT 201 qui reçoit une requête req_DNS d'un client 10 qui demande une donnée dont il est responsable, envoie au client la donnée resp_DNS en réponse à la requête.

5 Soit la donnée est dans le cache du nœud 201, soit le nœud 201 récupère la donnée sur un réseau Internet 50 par interrogation d'un serveur autoritaire non représenté. Pour une requête concernant une donnée dont il n'est pas responsable, le premier nœud 201 consulte une table d'identification DHT (non représentée sur la figure 1), appelée également table de routage, afin de
10 déterminer un deuxième nœud 204 responsable de la donnée à envoyer en réponse. Le premier nœud 201 interroge alors (DHT_req) le deuxième nœud 204. Le deuxième nœud 204 envoie au premier nœud 201 la donnée DHT_resp en réponse à la requête et le premier nœud 201 forme alors une réponse resp_DNS à destination du client qui a émis la requête et lui envoie.

15 Une telle architecture permet d'optimiser la répartition des données au sein des serveurs caches qui gèrent ainsi chacun un ensemble particulier de données et communiquent entre eux. Cependant, elle ne permet pas de répondre à tous les besoins inhérents à un système DNS.

En particulier, une telle architecture n'est pas adaptée pour traiter une
20 grande quantité de requêtes simultanées. En effet, lorsqu'un premier nœud DHT du réseau P2P reçoit une requête mais ne possède pas la réponse en cache, il cherche dans sa table un deuxième nœud approprié pour traiter la requête. Une fois ce deuxième nœud déterminé, le premier nœud lui retransmet la requête puis reste en attente d'une réponse à retransmettre au client. Ainsi,
25 le premier nœud qui reçoit la requête mais n'a pas la réponse à celle-ci doit rester en attente de la réponse. Il en résulte que, bien que les tables DHT soient adaptées pour gérer une grande quantité de données, une architecture DNS basée sur des tables DHT et des serveurs caches organisés en réseau P2P n'est pas adaptée pour supporter des charges importantes. En
30 conséquence, la Qualité de Service offerte par une telle architecture se dégrade lorsque la charge augmente puisque les délais pour répondre aux requêtes augmentent.

Un des buts de l'invention est de remédier à des insuffisances de l'état de la technique.

L'invention répond à ce besoin en proposant un système de traitement d'une requête émise par un client pour obtenir une donnée, comprenant :

- 5 - une pluralité de nœuds organisés en réseau P2P, adaptés pour stocker un ensemble de données réparties dans les différents nœuds et mémorisant une table d'identification adaptée pour identifier, celui parmi la pluralité de nœuds qui est apte à fournir la donnée en réponse à la requête, caractérisé en ce qu'il comprend :
 - 10 - un serveur mandataire mémorisant ladite table et auquel la requête est destinée à être adressée, ledit serveur étant adapté pour, suite à la réception de la requête, uniquement :
 - identifier, à partir de la table, le nœud apte à fournir la donnée en réponse à la requête, et
 - 15 - retransmettre la requête au nœud identifié.

De façon avantageuse, l'architecture selon l'invention comprend un serveur mandataire qui est dédié exclusivement au routage des requêtes qu'il reçoit. On rappelle ici que, dans l'architecture selon l'art antérieur basée sur les tables distribuées DHT, chaque nœud DHT est adapté pour stocker des données et pour router une requête qu'il reçoit vers le nœud DHT qui devrait héberger la donnée. Avec l'invention, la fonction de routage est externalisée et réalisée uniquement par le serveur mandataire. Or, cette fonction de routage est habituellement très coûteuse en temps dans l'architecture DHT classique de l'art antérieur. Grâce à l'invention, les nœuds DHT sont uniquement dédiés au stockage de données et s'affranchissent ainsi de la fonction, coûteuse, de routage.

On comprend qu'avec l'architecture selon l'invention, les différentes tâches à effectuer lors de résolutions DNS sont effectuées par des entités dédiées, contrairement aux solutions de l'art antérieur. Ainsi, le stockage proprement dit, est réalisé par les nœuds du réseau P2P, et le routage des requêtes des clients à l'intérieur de la plate-forme constituée des nœuds du

réseau P2P est exclusivement réalisé par le serveur mandataire. Le stockage de données par les nœuds correspond à la résolution DNS proprement dite.

En outre, avec l'architecture selon l'invention, les nœuds du réseau P2P, en l'espèce les serveurs caches DNS ne fonctionnent plus de manière indépendante, mais collaborent et gèrent chacun une partie des données mises en cache. Ainsi, le stockage des données sur les différents serveurs caches est optimisé, contrairement à une architecture DNS classique où des mêmes données peuvent être stockées sur plusieurs serveurs DNS caches. On évite ainsi une redondance au niveau des données stockées, ce qui est avantageux lorsque le nombre de données à stocker croît, et/ou lorsque la taille des données à stocker est importante. C'est le cas dans les systèmes DNSSEC.

L'architecture selon l'invention est évolutive en termes de nombre de nœuds réseau : elle peut s'adapter facilement à des augmentations de trafic DNS. Une évolution en termes de nombre de nœuds n'impacte pas les clients, par exemple par des messages supplémentaires. Ainsi, cela permet de définir une architecture comprenant un nombre prédéfini important de nœuds caches, chacun gérant peu de données mais offrant en contrepartie un accès mémoire rapide.

Avantageusement, avec le système selon l'invention, le nœud auquel la requête est retransmise est adapté pour envoyer la donnée directement au client.

L'architecture selon l'invention permet de limiter les échanges de messages suite à une requête d'un client. Ainsi, la requête est envoyée du client au serveur mandataire qui la route vers le serveur cache responsable de son traitement. De façon avantageuse, le serveur qui détient la donnée en cache envoie directement cette donnée au client. La réponse, contrairement à la requête, ne transite ainsi pas par l'intermédiaire du serveur mandataire.

Dans une réalisation de l'invention, le système comprend un serveur résolveur, apte à recevoir la requête du nœud lorsque la donnée en réponse n'est pas stockée sur ledit nœud, à effectuer une résolution auprès d'un serveur autoritaire de l'Internet pour obtenir la donnée, et à envoyer la donnée directement au client.

De façon avantageuse, le serveur résolveur adapté pour effectuer une résolution DNS en interrogeant un serveur autoritaire de l'internet transmet la donnée reçue du serveur autoritaire d'une part au nœud responsable pour mémorisation en cache, et d'autre part, directement au client en réponse à sa
5 requête. Ainsi, l'architecture selon l'invention optimise le traitement de la requête en évitant de faire transiter la donnée à envoyer en réponse au client par des nœuds intermédiaires.

Dans un exemple de réalisation du système selon l'invention, les nœuds sont des nœuds DNS caches et la requête est une requête de résolution de
10 nom de domaine.

Dans un exemple de réalisation du système selon l'invention, la table d'identification est une table DHT.

L'invention concerne aussi un réseau comportant le système selon l'invention et une pluralité de terminaux clients, lesdits terminaux étant
15 configurés pour envoyer des requêtes pour obtenir une donnée exclusivement au serveur mandataire.

L'invention concerne également un terminal pour le réseau selon l'invention configuré pour envoyer des requêtes pour obtenir une donnée exclusivement au serveur mandataire.

20 L'invention porte également sur un procédé de traitement d'une requête émise par un client pour obtenir une donnée, un ensemble de données étant réparties dans une pluralité de nœuds d'un réseau P2P mémorisant une table d'identification adaptée pour identifier, celui parmi la pluralité de nœuds qui est apte à fournir la donnée en réponse à la requête,

25 caractérisé en ce qu'il comprend :

- une étape de réception de la requête du client par un serveur mandataire mémorisant ladite table,
- une étape d'identification à partir de la table du nœud apte à fournir la donnée, et
- 30 - une étape de retransmission de la requête au nœud identifié.

L'invention concerne aussi un serveur mandataire comprenant :

- des moyens de réception d'une requête émise par un client pour obtenir une donnée,

- des moyens de mémorisation d'une table d'identification adaptée pour identifier, parmi une pluralité de nœuds organisés en réseau P2P, lesdits
5 nœuds étant adaptés pour stocker un ensemble de données réparties dans les différents nœuds, celui parmi la pluralité de nœuds qui est apte à fournir la donnée en réponse à la requête,

- des moyens d'identification, adaptés pour identifier à partir de la table le nœud apte à fournir la donnée, et

10 - des moyens de transmission, adaptés pour retransmettre la requête au nœud identifié.

L'invention concerne également un programme d'ordinateur sur un support de données et chargeable dans la mémoire interne d'un serveur, le programme comprenant des portions de code pour l'exécution des étapes du
15 procédé selon l'invention, lorsque le programme est exécuté sur ledit dispositif client.

Enfin, l'invention concerne un support de données sur lequel est enregistré le programme d'ordinateur selon l'invention.

20 De nombreux détails et avantages de l'invention seront mieux compris à la lecture de la description d'un mode particulier de réalisation en référence aux figures annexées données à titre non limitatif et dans lesquels :

- le figure 1 (déjà commentée) représente une architecture de traitement d'une requête d'un client pour obtenir une donnée de l'art antérieur ;

25 - la figure 2 représente une architecture, selon un premier exemple de réalisation de l'invention, adaptée pour traiter une requête de client en vue d'obtenir une donnée dans un premier où le nœud identifié détient la donnée à envoyer en réponse dans sa mémoire cache ;

- la figure 3 représente les échanges entre les différentes entités de
30 l'architecture de l'invention dans un second cas où le nœud identifié ne détient pas encore la donnée dans sa mémoire cache ;

- la figure 4 présente les étapes d'un procédé de traitement d'une requête d'un client selon un mode particulier de réalisation de l'invention ;

On rappelle au préalable que, dans l'architecture de l'art antérieur représentée sur la figure 1, une pluralité de serveurs caches 201, 202, 203, 204, 205 sont organisés en nœuds "DHT" (pour "Distributed Hash Table", ou table de condensats) d'un réseau "P2P" 20 (pour "Peer-to-Peer", ou réseau pair à pair). Chaque nœud DHT du réseau P2P 20 mémorise une table d'identification 200, ou table DHT, ou table de routage associée au réseau P2P 20 de serveurs caches. Seule la table d'identification 200 mémorisée dans le nœud DHT 201 est représentée. La même table d'identification 200 est stockée par chacun des nœuds. La table d'identification 200 permet d'identifier à partir d'une requête d'un client le nœud DHT responsable du traitement de la requête.

On notera d'emblée que, dans la présente description, on entend par traitement de la requête :

- dans le cas où le nœud DHT identifié possède en cache la donnée à envoyer en réponse à la requête, l'envoi de cette donnée de réponse au client, ou

- dans le cas où le nœud DHT identifié ne possède pas en cache la donnée à envoyer, l'obtention de la donnée à envoyer en réponse à la requête auprès d'un serveur résolveur, lequel est adapté pour faire une résolution DNS en interrogeant un serveur autoritaire de l'internet.

Le premier cas où le nœud DHT identifié possède déjà la donnée dans sa mémoire cache correspond au cas où cette donnée a déjà été demandée à ce nœud DHT par une requête précédente, envoyée par exemple par un autre client. Le deuxième cas où le nœud DHT identifié ne possède pas la donnée dans sa mémoire cache correspond au cas où la donnée n'a pas encore été demandée au nœud DHT. Cependant, le protocole DHT a déjà identifié ce nœud comme étant celui qui est présumé détenir la donnée à envoyer en réponse ou, en toute hypothèse, a pour rôle de l'obtenir.

La figure 2 est relative au premier cas et la figure 3 au deuxième cas.

La figure 2 représente une architecture, selon un premier exemple de réalisation de l'invention, adaptée pour traiter une requête de client en vue d'obtenir une donnée dans le premier cas mentionné ci-dessus.

Dans l'architecture selon la figure 2, un client 10 est adapté pour émettre
5 des requêtes "DNS" (pour "Domain Name Service", ou service de nom de domaine), notées req, en vue d'effectuer des résolutions de nom de domaine. Par exemple, avec une telle requête, le client demande l'adresse IP associée au nom "www.mabanque.fr", afin d'accéder au site internet de la banque. Le client 10 est configuré de telle sorte que ses requêtes DNS req sont
10 exclusivement adressées à un serveur mandataire 30. Par exemple, une configuration consiste à préciser l'adresse IP du serveur mandataire 30 dans un fichier de configuration propre au client.

L'architecture de la figure 2 diffère de celle de la figure 1 (art antérieur) notamment par le fait qu'elle comprend un serveur mandataire 30. Le serveur
15 mandataire 30 comprend :

- des moyens 301 de réception, adaptés pour recevoir les requêtes req émises par le client 10 ;
- des moyens de mémorisation 302, adaptés pour mémoriser la table d'identification 200 ;
- 20 - des moyens d'identification 303 adaptés pour identifier, à partir de la requête req reçue du client et de la table d'identification 200, le nœud DHT responsable du traitement de la requête ;
- des moyens de transmission 304, agencés pour transmettre la requête req reçue du client 10 au nœud DHT identifié pour
25 traiter la requête par les moyens d'identification 303.

Les moyens 302 de mémorisation sont par exemple une mémoire non volatile de type "EEPROM" (pour "Electrically Erasable Programmable Read Only Memory"). La table d'identification 200 est destinée à être importée par le serveur mandataire 30 depuis au moins un nœud DHT, comme expliqué plus
30 loin dans la description.

La table d'identification 200 comprend un ensemble d'index. Chaque index, plus précisément, une plage d'index identifie un nœud DHT de manière

unique. En outre, une requête de client permet de déterminer un index dans la table d'identification. Il est ainsi possible de déterminer à partir d'une requête de client, le nœud DHT responsable du traitement de cette requête. L'index associé à une requête de client est obtenu par application d'une fonction de hachage à la requête du client. Par exemple, la fonction SHA-1 (pour "Secure Hash Algorithm-1") peut être utilisée. L'invention n'est bien sûr pas limitée à la fonction SHA-1.

Ainsi, les moyens d'identification 303 du nœud 201 comprennent des moyens de calcul adaptés pour calculer un hash de la requête req reçue du client 10 et des moyens de recherche pour chercher dans la table d'identification 200 le nœud DHT en prenant comme index le hash de la requête.

Pour rappel, on suppose ici que le nœud identifié comme étant responsable du traitement de la requête req, par exemple le nœud DHT 201, détient la donnée à envoyer en réponse dans sa mémoire cache. Selon le premier exemple de réalisation de l'architecture selon l'invention décrit ici en relation avec la figure 2, le nœud 201 est adapté pour envoyer une réponse resp comprenant la donnée directement au client. Autrement dit, la réponse resp n'est pas transmise du nœud DHT 201 au serveur mandataire 30 qui a transmis la requête req au nœud 201. Elle est transmise du nœud DHT 201 au client ayant émis la requête, sans passer par l'intermédiaire du serveur mandataire 30 par lequel la requête a transité.

Un seul client 10 est représenté sur la figure 2. Bien sûr, l'invention n'est pas limitée au traitement de requêtes issues d'un seul client, mais est adaptée pour traiter des requêtes issues d'une pluralité de clients.

De même, un seul serveur mandataire 30 est représenté sur la figure 2. L'invention n'est pas limitée à un seul serveur mandataire. En particulier, pour garantir une bonne Qualité de Service en termes de temps de réponse, notamment dans des situations de charge importante, il est possible d'avoir une pluralité de serveurs mandataires vers lesquels des requêtes de clients sont dirigées vers différents serveurs mandataires. Disposer de plusieurs serveurs mandataires permet de limiter la charge de chacun des serveurs. Par exemple,

un équipement réseau, tel qu'un routeur peut être utilisé pour diriger les requêtes issues des clients vers les différents serveurs mandataires.

La table d'identification 200 stockée sur chaque serveur DHT du réseau P2P 20 est également mémorisée sur le serveur mandataire 30. Afin de
5 maintenir une cohérence entre la table mémorisée sur le serveur mandataire 30 et la table stockée sur les nœuds DHT, il est prévu un mécanisme d'importation de la table d'identification 200 par le serveur mandataire 30 depuis un nœud DHT. Par exemple, le serveur mandataire 30 émet régulièrement une requête de mise à jour de la table vers un nœud DHT proche géographiquement de lui.
10 Dans un autre exemple de réalisation, un nœud DHT peut être paramétré afin d'envoyer régulièrement la table d'identification 200 qu'il mémorise au serveur mandataire 30. Dans un autre exemple de réalisation, le nœud DHT peut également être paramétré pour n'envoyer une mise à jour de la table d'identification que lors d'une évolution de celle-ci. Par exemple, une telle
15 évolution de la table peut être nécessaire lorsqu'un nœud DHT devient hors service. Dans ce cas le protocole DHT utilisé est adapté pour recalculer la table DHT à partir des nœuds encore actifs.

La figure 3 représente le système selon l'invention, et plus précisément les échanges entre les entités du système, dans un second cas de figure. Ce
20 second cas de figure correspond au cas où le nœud DHT 201 responsable du traitement de la requête req envoyée par le client 10 ne détient pas encore la donnée en cache, le nœud DHT 201 est adapté pour obtenir la donnée à envoyer en réponse auprès d'un serveur résolveur 40. A cet effet, le nœud DHT 201 comprend des moyens d'interrogation (non représentés sur la figure 3) du
25 serveur résolveur 40. Le nœud DHT 201 transmet la requête req au serveur résolveur 40 qui est apte à effectuer une résolution DNS en interrogeant un serveur autoritaire DNS (non représenté sur la figure 3) à travers un réseau Internet 50. Le nœud 201 comprend également des moyens pour recevoir une réponse resp du serveur résolveur 40, en l'espèce la réponse à la requête req,
30 et pour la mémoriser dans sa mémoire cache. Dans cet exemple de réalisation, le serveur résolveur 40 est adapté pour envoyer la réponse resp directement au client qui a émis la requête. Avantagusement, dans ce mode de réalisation, la

réponse n'est pas envoyée par le nœud DHT au client, mais directement par le résolveur 40 au client. Ainsi, le temps de réponse est optimisé. Dans un cas particulier où la requête req est une requête DNSSEC, alors le serveur résolveur 40 est adapté pour recevoir outre la donnée réponse à la requête, une signature de ladite donnée. Il est adapté également pour vérifier ladite signature.

Dans les exemples d'architecture décrits en relation avec les figures 2 et 3 et dédiées à un service de résolution DNS, on estime que le nombre de nœuds du réseau P2P est fini et inférieur à une valeur prédéterminée, par exemple 300, et que ces nœuds sont opérés par un même opérateur. Le fait que tous les nœuds du réseau P2P soient opérés par le même opérateur facilite la communication entre nœuds et la gestion des nœuds. En outre, la valeur prédéterminée, considérée comme correspondant à un petit nombre de nœuds, permet à chaque nœud d'avoir connaissance de tous les autres nœuds, et conformément à la technologie DHT, permet à l'ensemble des nœuds du réseau de partager une même table d'identification. Gérer une seule table d'identification pour l'ensemble des nœuds du réseau P2P facilite l'exportation de la table vers le serveur mandataire et l'externalisation de la fonction de routage. En outre, cela optimise le routage des requêtes des clients puisqu'un nœud responsable du traitement d'une requête est immédiatement identifié par consultation de la table. Cela peut ne pas être le cas lorsqu'un réseau P2P est constitué de plusieurs milliers de nœuds. En effet, dans ce cas, plusieurs tables DHT peuvent être nécessaires pour stocker l'ensemble des informations réparties sur les nœuds du réseau P2P et trouver une information parmi ces nœuds peut nécessiter plusieurs bonds successifs d'un nœud à un autre.

Les étapes d'un procédé de traitement d'une requête selon un mode de réalisation particulier de l'invention vont maintenant être décrites en relation avec la figure 4.

Au préalable, on notera que le client 10 est paramétré pour que les requêtes qu'il émet soient dirigées vers le serveur mandataire 30.

Dans une étape initiale E0, le serveur mandataire 30 de la figure 2 reçoit du client 10 une requête req de résolution DNS.

5 Dans une étape E1 d'identification d'un nœud, le serveur mandataire 30 consulte la table d'identification 200 qu'il mémorise afin d'identifier le nœud DHT responsable du traitement de la requête req. Puis, dans une étape suivante E2 de retransmission, le serveur mandataire 30 retransmet la requête req au nœud DHT identifié comme étant responsable du traitement de la requête, par
10 exemple le nœud 201 de la figure 2.

Dans une étape E'3, le nœud 201 reçoit la requête req du serveur mandataire 30.

Dans une étape E'4 de traitement de la requête, le nœud 201 responsable du traitement de la requête recherche la donnée à envoyer en
15 réponse à la requête dans sa mémoire cache.

Dans le cas où le nœud 201 détient la donnée en cache, il envoie directement au client 10 selon la figure 2, la donnée dans un message resp.

Dans le cas où le nœud 201 ne possède pas la donnée en cache, alors dans une étape non représentée sur la figure 4, il envoie la requête req vers le
20 serveur résolveur 40 selon la figure 3 adapté pour effectuer une résolution DNS afin d'obtenir la réponse à la requête. A cette fin, le serveur résolveur 40 émet la requête req auprès d'un serveur autoritaire (non représenté) de l'Internet 50 selon la figure 3. Le serveur résolveur 40 qui a obtenu du serveur autoritaire la donnée en réponse à la requête req, transmet celle-ci, d'une part au nœud 201
25 qui la stocke dans sa mémoire cache, et d'autre part directement au client en réponse à la requête req. Dans le cas du système DNSSEC, le serveur résolveur reçoit du serveur autoritaire outre la donnée, une signature de la donnée permettant de vérifier l'intégrité de la donnée. Le serveur résolveur est adapté également pour vérifier la signature de la donnée.

30 Les modules 301, 302, 303 et 304 du serveur mandataire 30 selon la figure 2, ou la figure 3, sont agencés pour mettre en œuvre celles des étapes précédemment décrites du procédé de traitement d'une requête d'un client qui

sont mises en œuvre par le serveur mandataire 30. Il s'agit de préférence de modules logiciels comprenant des instructions logicielles pour faire exécuter les étapes du procédé de traitement d'une requête d'un client par le serveur mandataire.

5 L'invention concerne donc aussi :

- un programme d'ordinateur comportant des instructions pour la mise en œuvre du procédé de traitement d'une requête d'un client tel que décrit précédemment lorsque ce programme est exécuté par un processeur ;

- un support d'enregistrement lisible par un serveur mandataire sur lequel
10 est enregistré le programme d'ordinateur décrit ci-dessus.

Les modules logiciels peuvent être stockés dans, ou transmis par un support de données. Celui-ci peut être un support matériel de stockage, par exemple un CD-ROM, une disquette magnétique ou un disque dur, ou bien un support de transmission tel qu'un signal, ou un réseau de télécommunication.

15

L'invention est décrite ici dans le cas du traitement de requêtes DNS. L'invention n'est bien sûr pas limitée au traitement de requêtes DNS et peut s'appliquer à tout protocole basé sur l'envoi de requêtes pour obtenir une donnée.

20

REVENDICATIONS

1. Système de traitement d'une requête émise par un client pour obtenir une donnée, comprenant :

- 5 - une pluralité de nœuds (201, 202, 203, 204, 205) organisés en réseau P2P, adaptés pour stocker un ensemble de données réparties dans les différents nœuds et mémorisant une table d'identification (200) adaptée pour identifier, celui parmi la pluralité de nœuds qui est apte à fournir la donnée en réponse à la requête,
- 10 caractérisé en ce qu'il comprend :
- un serveur mandataire (30) mémorisant ladite table et auquel la requête est destinée à être adressée, ledit serveur étant adapté pour, suite à la réception de la requête, uniquement :
- identifier, à partir de la table, le nœud apte à fournir la donnée en
- 15 réponse à la requête, et
- retransmettre la requête au nœud identifié.

2. système selon la revendication 1, dans lequel le nœud auquel la requête est retransmise est adapté pour envoyer la donnée directement au

20 client.

3. Système selon la revendication 1, comprenant un serveur résolveur, apte à recevoir la requête du nœud lorsque la donnée en réponse n'est pas stockée sur ledit nœud, à effectuer une résolution auprès d'un serveur

25 autoritaire de l'Internet pour obtenir la donnée, et à envoyer la donnée directement au client.

4. Système selon la revendication 1, dans lequel les nœuds sont des nœuds DNS caches et la requête est une requête de résolution de nom de

30 domaine.

5. Système selon la revendication 1, dans lequel la table est une table DHT.

6. Réseau comportant le système selon la revendication 1 et une pluralité
5 de terminaux clients, lesdits terminaux étant configurés pour envoyer des requêtes pour obtenir une donnée exclusivement au serveur mandataire.

7. Terminal pour le réseau selon la revendication 6 configuré pour
envoyer des requêtes pour obtenir une donnée exclusivement au serveur
10 mandataire.

8. Procédé de traitement d'une requête émise par un client pour obtenir
une donnée, un ensemble de données étant réparties dans une pluralité de
nœuds d'un réseau P2P mémorisant une table d'identification adaptée pour
15 identifier, celui parmi la pluralité de nœuds qui est apte à fournir la donnée en
réponse à la requête,

caractérisé en ce qu'il comprend :

- une étape (E0) de réception de la requête du client par un serveur
mandataire mémorisant ladite table,
- 20 - une étape (E1) d'identification à partir de la table du nœud apte à fournir
la donnée, et
- une étape (E2) de retransmission de la requête au nœud identifié.

9. Serveur mandataire (30) comprenant :

- 25 - des moyens (301) de réception d'une requête émise par un client pour
obtenir une donnée,
- des moyens (302) de mémorisation d'une table d'identification (200)
adaptée pour identifier, parmi une pluralité de nœuds organisés en réseau P2P,
lesdits nœuds étant adaptés pour stocker un ensemble de données réparties
30 dans les différents nœuds, celui parmi la pluralité de nœuds qui est apte à
fournir la donnée en réponse à la requête,

- des moyens (303) d'identification, adaptés pour identifier à partir de la table le nœud apte à fournir la donnée, et
- des moyens (304) de transmission, adaptés pour retransmettre la requête au nœud identifié.

5

10. Programme d'ordinateur sur un support de données et chargeable dans la mémoire interne d'un serveur, le programme comprenant des portions de code pour l'exécution des étapes du procédé selon la revendication 8, lorsque le programme est exécuté sur ledit dispositif client.

10

11. Support de données sur lequel est enregistré le programme d'ordinateur selon la revendication 10.

15

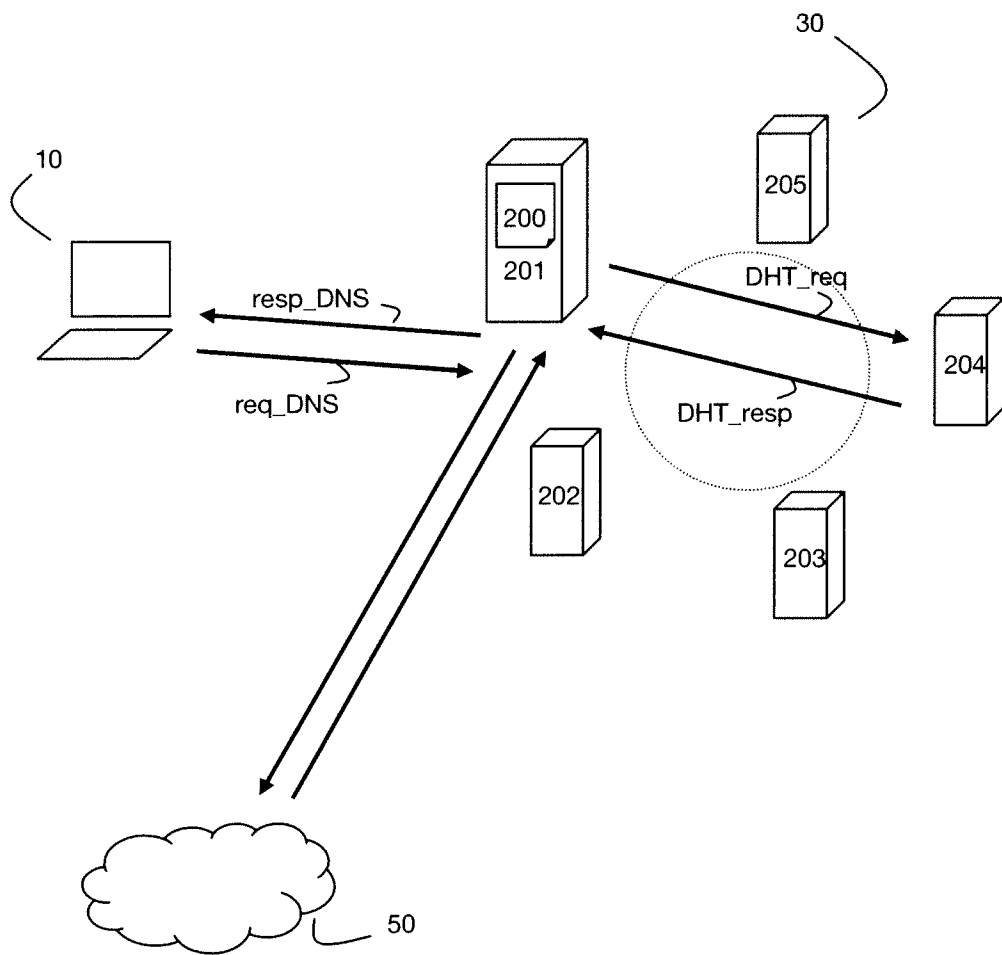


Figure 1

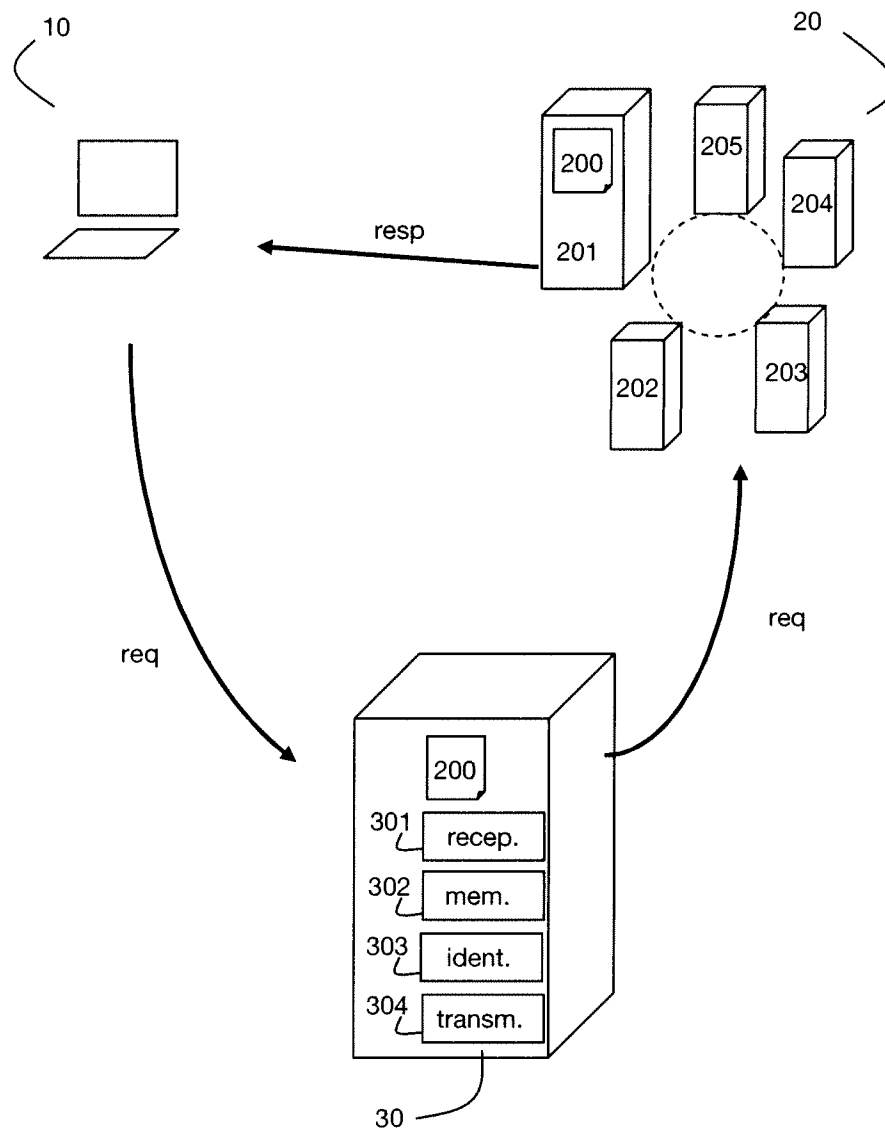


Figure 2

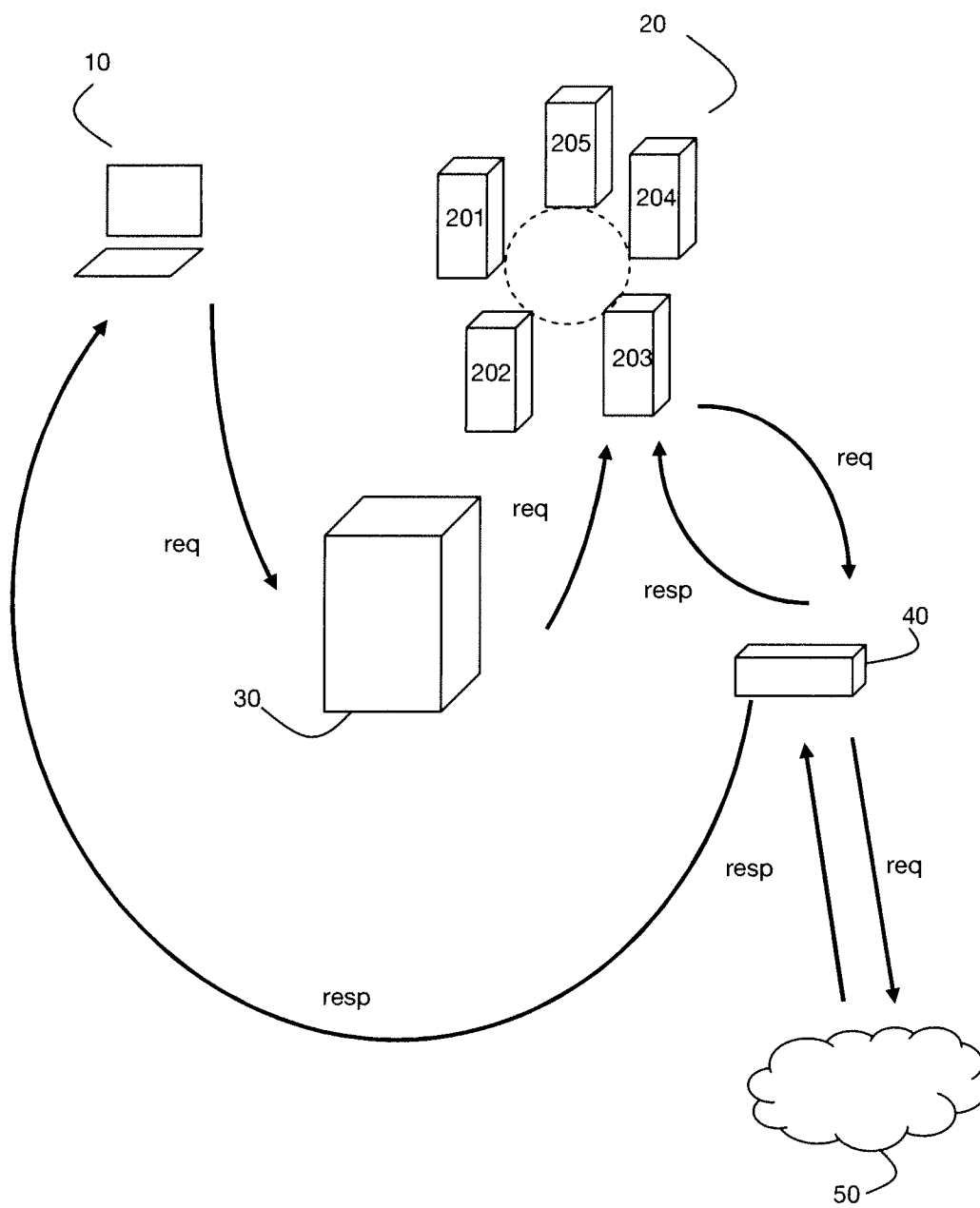


Figure 3

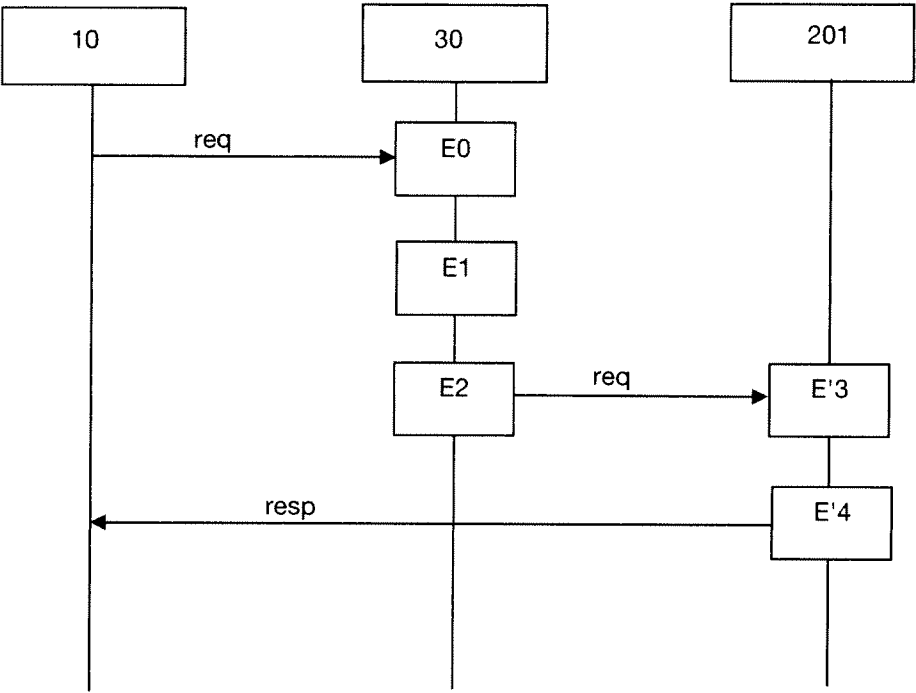


Figure 4