SAN JUAN – Comment ça marche : notions fondamentales sur le DNS
Lundi 12 mars 2018 – 17h00 à 18h30 AST
ICANN61 | San Juan, Porto Rico

CATHY PETERSEN : Bonjour à tous. Soyez les bienvenus à la séance sur les fonctionnements du DNS, les aspects fondamentaux. Nous avons monsieur Larson qui est le vice-président et qui est responsable de tout ce qui est recherche dans le bureau du PDG de l'ICANN. Comme vous pouvez le voir, vous êtes peu nombreux au point de vue physique ; venez à l'avant dans les premiers rangs. Et nous allons entamer notre séance.

MATT LARSON : Donc bonjour à tous. Soyez les bienvenus dans cette séance sur le fonctionnement et les explications sur le fonctionnement élémentaire du DNS. Nous sommes très peu nombreux. Si vous avez des questions, levez la main, on peut s'arrêter quand on voudra. Et nous avons beaucoup de matériel donc nous pourrons répondre à toutes les questions.

Les adresses IP sont faciles à comprendre pour les machines mais difficiles à retenir pour les personnes. Voilà pourquoi nous allons parler du DNS.

Lorsque nous avions des adresses IPv4, l'espoir existait que les hommes pourraient se rappeler de deux ou trois adresses IPv4. Mais ce que vous voyez là, c'est une adresse IPv6 qui a très peu de caractères. Il y a des adresses beaucoup plus longues et il est vraiment impossible de se rappeler les adresses IPv6. Les gens ont besoin d'utiliser des noms. Les ordinateurs se servent des numéros.

Lorsque l'internet a commencé, les noms étaient faciles. Nous avions ce que l'on appelait les étiquettes uniques. C'était des domaines qui n'existaient pas encore. C'était des étiquettes très courtes avec 24 caractères maximum. Tous les noms de l'internet devaient pouvoir rentrer dans un espace de nom à 24 caractères. On les appelait les noms de l'hôte. Hôte, cela signifie en fait ordinateur.

Lorsqu'on recherchait des adresses IP pour que les hommes puissent utiliser des noms, utiliser un ordinateur, c'est ce que l'on appelle la résolution des noms. Et avant l'existence du DNS, on utilisait des fichiers hôtes appelés host.txt. Cela a la même fonction mais avec un format légèrement différent des nouveaux fichiers host. C'est un fichier de texte avec des noms et des adresses et des noms d'hôtes et les adresses IP y afférentes.

Ceci est centralisé par NIC, le centre d'information des réseaux, qui a un contrat passé avec le gouvernement des États-Unis

pour qu'il réalise des tâches d'administration du réseau. L'une d'elles consiste à maintenir les fichiers hôtes. À l'époque, l'internet était beaucoup plus petit. À l'époque, cela s'appelait ARPANET et c'était une expérience menée par le département de la Défense des États-Unis. Il y avait à l'époque beaucoup moins d'hôtes qu'à l'heure actuelle. Aujourd'hui, il y a des dizaines de milliers. Mais à l'époque, il était raisonnable de tenir tout centralisé.

La communication se faisait par courriels avec un gestionnaire de réseau qui modifiait le nom ou l'adresse IP. Il envoyait un courriel à NIC pour lui dire : « Écoutez, j'ai fait cela. Changez le nom ou l'adresse IP de cette adresse et mettez ceci. » NIC, donc, avait toujours cette copie mère ou centrale de ce fichier et il en faisait une nouvelle copie une fois par semaine. Et les gestionnaires de réseau, lorsqu'ils décidaient que leur fichier devait être mis à jour, ils téléchargeaient un nouveau fichier TXT. Les choses ne changeaient pas trop souvent et on téléchargeait tout cela en utilisant le FTP qui utilisait très peu de ressources technologiques mais qui présentait certains problèmes prévisibles, si on y réfléchissait un tout petit peu.

L'un des problèmes concernait la controverse ou le conflit des noms. Si nous avons 24 caractères pour définir le nom de l'ordinateur et que la taille du réseau s'accroît, plus il y a des

dispositifs sur les réseaux, plus il y a de conflits, donc il y a plus de noms et il est beaucoup plus difficile d'éviter les doublons.

Et pour compliquer les choses, le maintien de ce fichier se faisait très simplement. Avec NIC, on travaillait avec un éditeur de texte. Il n'y avait pas de base de données. C'était un fichier de texte. Il n'avait pas moyen d'éviter les doublons et de temps en temps, on voyait apparaître des doublons.

Un autre problème classique était celui de la synchronisation. Toutes les personnes n'avaient pas toujours la même version du fichier. Et la charge de trafic et du téléchargement était difficile aussi. Le fichier a commencé à prendre tellement de complexité que la largeur de bande était compliquée. Donc à l'époque, avant que je ne grandisse, on m'a dit qu'à l'époque, le téléchargement du fichier prenait plus longtemps que le fait de mettre à jour ce fichier. On ne pouvait jamais avoir la dernière version du fichier parce que le téléchargement prenait plus longtemps que la mise à jour.

Donc on ne pouvait pas continuer à fonctionner de la sorte. Au début des années 80, on a commencé à se demander comment on pouvait remplacer host.txt. En premier lieu, il fallait résoudre les questions d'échelle. Et puis on a réfléchi au premier sujet, le problème de l'extensibilité, mais quand nous avions des

problèmes de routage des courriels. Le résultat est celui des noms de domaine, le DNS.

Voilà le résumé du DNS. Il y a une seule manière de résumer le DNS. Le DNS est une base de données distribuée. Et dans cette base de données, les données sont maintenues au niveau local. Tout le monde a sa propre partie de la base de données et s'occupe de la maintenance de ses propres données. Mais ces données sont disponibles au niveau mondial parce que cette base de données est distribuée partout dans le monde. Donc on maintient ces données au niveau local mais on peut voir les données de tout le monde.

Les fichiers du DNS ont un modèle client-serveur. Les résolveurs sont du côté clients, ils envoient des consultations. Ce que nous avons dit, c'est que les serveurs de noms répondent à des questions. Donc le résolveur envoie les questions et les serveurs de noms y répondent.

Le DNS utilisent une mémoire cachée pour améliorer la performance et cela signifie que, comme nous parlons d'une base de données distribuée partout dans le monde, la vitesse de la lumière est [une seule définie]. Lorsqu'on fait une recherche dans notre base de données, il faut peut-être demander sur plusieurs bases de données ; cela prend un certain temps. Et cela est très utile. Il est évidemment vital de se rappeler le

dernier résultat de recherche mais tous les résultats intermédiaires de cette recherche. Cela va dans la mémoire cachée et accélère le processus pour la prochaine fois.

Le DNS utilise la réplication pour offrir la redondance et la distribution des charges. Comme je l'ai dit, tout le monde maintient sa propre copie de la base de données mais la réplication est importante, c'est-à-dire il y a des copies multiples de chaque base de données. Si nous avions une copie de notre base de données et que cela se perdait, personne ne pourrait chercher quoi que ce soit. Nous avons des copies multiples. Cela signifie que nous avons une redondance, ce qui distribue la charge. Si beaucoup de personnes font des enquêtes ou des recherches, la charge de distribution se diffuse entre les différentes copies.

Nous avons ici différents composants sur une seule diapositive, et nous allons aborder la question des éléments en détail. Mais la vue d'ensemble nous aide à voir ce que nous visons. Nous allons commencer à gauche, en bas. À gauche, en bas, nous avons un dispositif connecté à l'internet, un téléphone par exemple un portable, mais tous les dispositifs connectés à l'internet doivent changer les noms en adresses. Tous les dispositifs qui utilisent le DNS ont un client appelé le résolveur stub, qui est une espèce de pont entre une application... par

exemple pour ce cas un navigateur, le résolveur est un pont entre l'application navigateur et le reste du DNS.

Alors le résolveur stub prend la demande d'une application, il transforme un nom en une adresse et puis il fait une quête au DNS qui va de l'avant et qui envoie cela à un résolveur récursif. Ce résolveur accepte une demande d'une application et transforme cette consultation pour le DNS, et il envoie cette demande et il attend une réponse. Le résolveur a, de l'autre côté, une fonction ou un rôle plus complexe. Il sait comment se mettre en contact avec différents systèmes de noms ou résolveurs faisant autorité.

Il se peut qu'il doive se mettre en contact avec plusieurs serveurs et ce serveur faisant autorité dise : « Moi, je n'ai pas la réponse mais je peux vous renvoyer à un second serveur. », qui vous dit : « Je n'ai pas la réponse mais je peux vous envoyer vers un autre troisième serveur qui est plus près. »

Donc ce résolveur récursif est intelligent et il surf sur ces serveurs jusqu'au moment où il trouve la réponse. Si nous cherchons dans le résolveur récursif, nous voyons qu'il a un résolveur de noms, et répond à des consultations qui sont présentées par le résolveur stub. Les résolveurs envoient des quêtes aux serveurs de noms qui font autorité.

Et nous avons vu qu'il y a aussi le cache. Tout ce que reçoit le résolveur du serveur faisant autorité le met dans la mémoire cachée pour l'utiliser plus tard pour des consultations ultérieures. Voilà donc l'écosystème du DNS décrit d'une manière très générale.

Je vais définir quelques points importants et quelques termes importants du DNS. En premier lieu, nous avons l'espace des noms. J'ai dit que le DNS est une base de données distribuée et sa structure est ce nous appelons l'espace des noms. Vous connaissez peut-être cela, quand on parle d'une base de données, vous pensez peut-être à une base de données relationnelle. La structure de ces bases de données comprend des tableaux multiples et ces tableaux ont des fils et dans chaque fil, nous avons des colonnes d'informations. Voilà comment ces bases de données relationnelles sont structurées.

Les bases de données du DNS sont différentes. C'est ce que nous appelons un arbre inversé. Là, vous avez un exemple avec une partie très petite de l'espace des noms du DNS. Dans un arbre inversé, la racine se trouve dans le haut et les branches poussent vers le bas.

C'est un arbre informatique et il n'est pas bizarre que les arbres informatiques soient avec la racine en haut et les branches en bas. Et tous les nœuds de cet arbre, tous ces carrés que vous

voyez là, ces rectangles que vous connaissez, ont une étiquette. Le nœud racine est très particulier. Il se trouve en haut. Il n'a pas d'étiquette ou son étiquette est l'étiquette vide, « nul ». Nous avons représenté cela parfois avec un point entre guillemets pour montrer qu'il n'y a rien, que cela est vide. Voilà donc l'espace des noms.

Bien des fois, nous appelons ces nœuds selon leur position et leur rapport à la racine. À gauche, nous avons les nœuds de premier niveau, puis nous avons ceux de second niveau et ainsi de suite. Nous allons vers le bas dans l'espace des noms. Et parfois, les nœuds sont appelés avec les noms de famille des pères et des enfants.

Vous avez ici la racine qui serait le père de .com et .com serait le père de example mais example est le fils de .com. Nous parlons de rapports père-enfant ou père-fils. Chacune de ces étiquettes à une série de caractères qui peuvent être utilisés. Nous les appelons LTH pour lettres, chiffres et traits d'union. Ce sont les seuls caractères que l'on peut utiliser dans les étiquettes de noms de DNS et la longueur maximale est de 63 caractères. Peu importe si ces étiquettes ont leur nom écrit en majuscules ou en minuscules, nous pouvons les mélanger mais cela va fonctionner de toute façon.

Chaque node a un nom de domaine et l'objectif de ce nom de domaine est de savoir où se trouve ce nœud dans l'espace des noms. Nous avançons vers la racine et nous écrivons le nom des étiquettes et au milieu, nous mettons le point. Vous voyez, le nœud que nous écrivons, c'est .com et puis nous avons encore .com et puis nous parvenons à la racine.

Il y a des noms spéciaux que nous appelons des noms de domaine pleinement qualifiés, des noms de domaine qui ne provoquent pas d'équivoque, où on sait parfaitement bien où se trouve un nom de domaine pleinement qualifié et finit par un point. C'est ce qui sépare le TLD .com de la racine. Nous sommes dans un domaine de premier niveau, nous mettons un point, nous mettons l'étiquette de la racine qui n'existe pas en fait. Donc cette étiquette se termine par un point et cela vous aide à comprendre. Et je pense que cela vous fait penser à des fichiers Windows ou à des fichiers Unix. C'est un exemple de structure de données qui peut être présenté comme un arbre inversé dans le système de fichiers qui représente des fichiers ou des répertoires. Et au lieu de cela, nous avons une route qui nous dit où se trouve le fichier ou le répertoire dans un directoire ainsi que le nom de domaine nous dit où se trouve un nom de domaine dans l'espace des noms.

Il y a aussi un autre terme important. La définition de domaine est une définition simple. C'est un nœud dans l'espace des noms

et tout ce qui est au-dessous. Ce que j'ai ici, c'est le domaine .com. Alors .com, ce serait ce qui se trouve en dessous. À vrai dire, c'est 131 millions. Il y en a beaucoup qui ne sont pas dans ces diapositives. Mais le domaine .com est quelque chose de très grand. C'est tout ce qui est en dessous de .com

Je vais comparer ceci maintenant avec le terme zone, et c'est un terme très important parce qu'il est mentionné très souvent et nous devons comprendre de quoi il s'agit. Rappelez-vous que nous sommes arrivés ici et que le DNS existe parce que nous avions besoin d'une base de données distribuée et centrale de l'information. Il a fallu donc chercher un système distribué pour que tout le monde puisse maintenir son information sur les adresses et sur les noms. Alors l'espace des noms est divisé pour qu'il puisse y avoir cette gestion distribuée. Et ces divisions administratives s'appellent zones.

Il y a différentes zones et chacun peut agir au sein de sa zone sans porter atteinte aux autres. Chacun gère sa propre zone et s'occupe de la maintenir. Ces zones sont obtenues par délégation depuis le point le plus élevé. Donc la zone de délégation s'appelle père et la zone créée, c'est la zone du fils, qui commence dans la racine et qui avance vers le bas. Je vais vous donner un autre exemple.

Voilà l'espace des noms encore une fois. Et si vous le regardez comme cela, nous n'avons pas suffisamment d'informations pour savoir où se trouvent les limites des zones. Donc l'espace des noms, si nous le regardons tout simplement, nous ne savons pas très bien comment il est divisé aux fins administratives.

Je vais vous en donner un exemple. Imaginez que vous regardez depuis un satellite ou d'une station spatiale et vous voyez l'Amérique du Nord. Si vous regardez l'Amérique du Nord du haut, vous ne savez pas exactement qu'il y a en fait trois pays là : le Canada, les États-Unis et le Mexique. Le satellite vous permet de voir tout cela mais vous ne saurez jamais quelles sont les limites administratives politiques parce qu'elles ne sont pas visibles. Nous avons besoin d'une information supplémentaire.

Nous voyons la même chose pour les zones. Nous voyons l'espace des noms mais à moins que nous ne sachions où se trouve la délégation, nous ne savons pas où est la délégation. Je sais où se trouvent les délégations dans l'espace des noms et c'est pour cela que je peux vous montrer les limites.

Nous avons au-dessus de tout la zone racine qui fait la délégation pour les zones qui sont au premier niveau. Nous disons qu'une zone possède l'information de délégation qui vise la zone à laquelle elle délègue. Ce processus de délégation se poursuit vers le bas dans l'espace des noms. La zone racine

délègue vers ce .com ; .com, dérive dans ce cas sur example.com et puis il y aura une délégation. Donc il faut penser à la taille de ces zones.

Commençons par la zone racine. La dernière fois où j'ai cherché, il y avait 1543 zones de domaines de premier niveau. C'est une zone assez petite. Elle a besoin d'informations de délégation sur 1543 zones qui sont au-dessous d'elle.

Mais si nous abordons .com par exemple, si nous considérons le matériel inclus par Verisign dans les sacs que vous avez reçus lors de votre enregistrement, ils disent qu'il y a 131 millions de noms dans la zone .com. C'est la plus grande qui existe, cette zone .com Donc elle doit avoir de l'information qu'elle délègue sur 131 millions de zones qui sont au-dessous de .com. Et la délégation peut continuer au-dessous de ce second niveau. On ne le voit pas sur la diapositive mais il peut y avoir une délégation du second niveau au troisième niveau quand on continue à avancer dans l'espace des noms.

Vous vous souvenez sans doute... [audio anglais]

Les noms correspondent à des consultations, un serveur de noms est autorisé pour une zone si on a la connaissance complète de la zone. Et le serveur de noms autoritaire ou faisant autorité peut répondre aux consultations si on fait ces consultations : « On m'a demandé quelque chose sur la zone,

voici l'information. » Et on peut me dire : « Vous avez fait une consultation, vous avez demandé un nom qui n'existe pas alors je peux vous dire que ce nom n'existe pas dans la zone. »

Les zones devraient avoir de nombreux serveurs faisant autorité. Comme je l'ai dit, ceci fournit de la redondance et distribue la charge. Alors chacune des zones a au moins un serveur de noms faisant autorité. Et en réalité, il y en a plus d'un. C'est bien d'avoir deux serveurs faisant autorité parce que s'il y en avait qu'un seul et qu'un problème se posait, on ne pourrait pas résoudre la question.

Si on avait de multiples serveurs faisant autorité, il est nécessaire de maintenir l'information synchronisée. L'idée est que l'information sur une zone devrait être la même dans tous les serveurs faisant autorité. Alors comment le faire ?

Le protocole a une forte synchronisation des données de zone dans les serveurs faisant autorité. Il y a un processus qui s'appelle transfert de zone qui permet de bouger dans la zone. Alors on désigne un serveur faisant autorité et cela nous permet de faire des changements dans la zone. Puis on a d'autres serveurs faisant autorité secondaires ou esclaves, c'est la même chose, et ces serveurs secondaires chargent ou collectent les données et font le transfert de zone et copient la zone du primaire au secondaire.

Alors il est important de signaler que tous les serveurs faisant autorité sont pareils. Ils ont les mêmes données. La seule différence, c'est le primaire et le secondaire, à savoir d'où ils collectent les données. Le primaire charge les données de zone depuis le disque et l'autre, depuis le primaire. Mais les données sont les mêmes. Mais bien sûr, il se peut qu'ils ne soient pas synchronisés à un moment donné parce qu'on peut faire un changement dans le serveur primaire mais ensuite, les données sont synchronisées. Cela est incorporé au protocole du DNS et vous ne devez pas vous préoccuper de la synchronisation des serveurs de noms parce que ceci a lieu tout simplement. Vous n'avez pas à le faire ; cela se fait tout seul, disons.

On a parlé de la zone. Maintenant, on va voir ce qui se passe avec les données dans la zone. Souvenez-vous des nœuds de l'espace de noms. Chacune de ces cases, disons, a un nom. Et la manière de le voir est la suivante. Un nom de domaine peut avoir différents types d'informations de données associées. L'adresse la plus commune, c'est une adresse IP. Alors on peut établir un lien entre une adresse IP et un nom de domaine.

Ce type d'informations qui accompagne un nom de domaine s'appelle registres de ressources. Il y a différents types de registres de ressources qui vont stocker différents types de données, normalement des adresses IPv4 et IPv6. Il y a différents types de registres mais il y a aussi d'autres types de données qui

sont stockées. Une zone a des registres de ressources et le fichier s'appelle fichier de zone. Il y a un fichier de zone pour chacune des zones et ils ne sont jamais mélangés. Un zone est tout simplement un ensemble de tous les registres de ressources.

Je vais vous montrer de quoi on parle. Il y a une manière standard de les écrire avec du texte. Les registres de ressources ont cinq champs. On ne va pas les voir en entier. Le message important ici, c'est que les registres de ressources peuvent être différents et il faut le transfert de données différentes. Voici les registres de ressources les plus usuels. Tout d'abord, IPv4 et IPv6, qui s'appelle le registre des adresses. Puis on a celui qui stock les adresses IPv6 AAAA, puis d'autres types qui sont mentionnés dans cette liste, qui incluent les types de ressources les plus usuelles.

Mais il y a bien d'autres registres de ressources. J'ai vu 84 types différents. Il y a un registre de l'IANA qui a le nom que vous voyez ici sur l'écran. Et vous allez trouver ce que vous voulez sur la diapositive. Sur la base de la taille du champ de registres, on peut en arriver jusqu'à 65 000.

Si nous pensons à quelque chose de tout neuf pour stocker dans le DNS, on peut penser à l'IFT. Il faut convaincre les gens que notre idée doit se transformer dans un nouveau type de DNS. Et

à ce moment-là, on va le créer, on va octroyer ou attribuer une classe dans le registre. Et les gens font continuellement ceci. Pas si fréquemment, il n'y en a que 84. Mais les gens créent au fur et à mesure de nouvelles choses que l'on peut introduire dans le DNS.

Mais de loin, la classe de données ou le type de données les plus usuelles sont les adresses. L'utilisation la plus usuelle du DNS, c'est pour faire le mapping des noms de domaine avec des adresses. Et voici les deux types de registres que je vous ai montrés auparavant. Ici, on a la représentation en texte d'un registre d'adresses et d'un registre AAAA. On a example.com puis le type adresse puis l'adresse elle-même. Voici donc un exemple d'un registre de ressources qui sera dans le fichier de zone pour l'example.com et .com a cette adresse IP. Et en bas, nous avons le AAAA et on a example AAAA et l'adresse IPv6. On a donc des registres A et des registres AAAA. C'est la manière la plus facile de faire le mapping des registres d'adresses.

Il y a d'autres manières de faire les registres. Ce qui est important, c'est que la plupart de ces classes sont utilisées par des gens qui consomment des informations du DNS, qui cherchent des informations dans le DNS parce qu'ils ont besoin de se connecter dans un serveur web, le navigateur a besoin de trouver quelque chose. Mais bien des fois, il y a des types qui sont utilisés par le DNS lui-même. Un exemple en est le registre

DNS et le registre NS et le registre SOA. Ceci n'intéresse personne sauf le DNS lui-même. Ce qui est intéressant, c'est que ce sont des types comme les autres types.

Moi, j'aime bien utiliser l'analogie d'un entrepôt. Imaginez, on va louer un entrepôt parce qu'on veut stocker X choses. On ne prend pas le camion et on commence à tout jeter dans le dépôt ; il faut l'organiser, il faut avoir les meubles suffisants, il faut mettre les boîtes d'une manière ordonnée. Alors on peut penser aux registres NS, on peut penser aux étagères. Mais personne en dehors du DNS est préoccupé par les autres types de registres, comme A ou AAAA.

Nous parlons des registres NS. C'est la manière dont nous disons les serveurs de noms faisant autorité pour une zone. Cet exemple montre deux registres NS. Ces registres nous disent ce qui suit : example.com, cette zone-là a deux serveurs de noms faisant autorité, le premier NS1 et l'autre, NS2. À gauche, on a le nom de la zone et à droite, on a le nom du serveur de noms.

Les registres NS sont un peu complexes parce qu'ils apparaissent dans deux sites en réalité : dans la zone père et dans la zone fils. Dans cette case, j'ai toute la liste de registres NS pour la zone .com. Il y a 13 serveurs de noms faisant autorité pour cette zone qui s'appellent, comme vous pouvez le voir, A.gTLD-servers.net, B... etc. Cette liste de registres, ces 13

registres apparaissent sur deux sites. Ils apparaissent dans la zone racine. Dans ce cas, la zone racine, c'est la zone paire. Ces registres NS de la zone paire sont ceux qui font la délégation. On dit au reste du DNS qu'en bas de la zone racine, nous avons la zone .com et la liste des registres NS apparaît encore une fois dans la zone qui, dans ce cas, s'appelle .com. Les registres NS.com apparaissent dans la racine : dans le père et puis dans .com encore une fois. Et quand on parle de la manière de chercher ceci dans le DNS, vous verrez comment il est important que ces registres NS se trouvent dans la zone paire.

Et je vais vous dire maintenant que la manière de faire les résolutions dans le DNS est de commencer dans la racine et suivant la délégation en cherchant les registres NS. Alors si on cherche quelque chose qui est en dessous de .com, on commence par la racine, on fait la délégation à .com, on va dans le serveur de noms de .com, on cherche la délégation qui va en dessous et ainsi de suite. On reviendra sur ce point-là. Merci.

Les registres NS incluent les noms seulement. Alors si on cherche l'exemple que je vous ai donné auparavant, nous voyons qu'example.com, un des serveurs, c'est NS1.example.com mais ce n'est pas suffisant parce qu'après, on a besoin de l'adresse IP de NS1.example.com pour pouvoir le contacter. Alors la délégation doit inclure également des registres d'adresses dans certains cas. Et ceci, on les appelle des

registres glue. C'est le registre d'adresses pour un serveur de noms.

Il y a un autre registre qui se trouve dans toutes les zones dénommé registre SOA. Je ne vais pas trop en parler. Je veux tout simplement signaler qu'il existe, qu'il y a un registre SOA par zone. C'est la partie supérieure de ce que l'on appelle apex ou la partie supérieure de la zone. Et la plupart des valeurs ont trait au transfert dont je vous ai parlé auparavant. On dit au serveur faisant autorité la fréquence de synchronisation de la zone.

Nous revenons maintenant au deuxième objectif du DNS. Le deuxième problème que le DNS a à résoudre, le routage du courrier. Le problème à résoudre, c'était comment faire parvenir le courrier sur la base d'une adresse email.

Par le passé, avant l'existence du DNS, il y avait des adresses email, à savoir utilisateur@ et le nom de l'hôte. Ces noms avaient 24 caractères au maximum. Voilà, c'était les directions courriel. Et avant le DNS, ce que cela voulait dire, c'est que mon courrier électronique allait dans le nom de l'hôte qui était à droite de l'adresse courriel. Il n'y avait pas moyen de dire mon adresse courriel est [inintelligible] mais le message, il faut l'envoyer à une autre boîte. Non. Si mon adresse était matt@foo, alors il fallait arriver à ce serveur. L'idée, c'était de séparer. C'est

l'attache du DNS. Alors voilà mon adresse et voilà l'adresse où ce courriel doit arriver. Alors le DNS offre la flexibilité nécessaire.

Nous avons un registre qui s'appelle le registre d'échange de courrier et en voici un exemple. Ces registres MX pour le cas d'example.com par exemple, ce registre indique où doit aller le courriel. Alors pour n'importe quel nom d'utilisateur@example.com, le registre MX dit que ce mail doit arriver dans une machine qui s'appelle mail.example.com Il y a des valeurs de référence, 10 et 20, et ceci va un peu à l'encontre de l'intuition parce qu'il y en a un qui est plus souhaitable que l'autre. Alors dans le cas de n'importe quelle adresse courriel @.com, il faut l'envoyer à mail.example.com. Mais si on ne peut pas le faire pour une raison quelconque, il faut essayer de l'envoyer à mailbackup.example.com. Alors il faut identifier les registres MX pour le nom de domaine. Il y a une référence entre le serveur de nom et le SMTP. Alors on cherche le registre MX pour l'adresse courriel et alors, le système sait où envoyer le message.

Jusqu'à présent, nous avons parlé de faire le mapping des noms avec les adresses IP ; c'est une tâche importante. Mais bien des fois, nous voulons faire l'inverse. Le mapping de noms IP s'appelle mapping vers l'avant. Mais que se passe-t-il si on veut faire le mapping IP ou non ? Et dans certains cas, c'est nécessaire.

Imaginez par exemple qu'il y a un réseau qui veut détecter un problème et qui veut voir quels sont les routeurs entre nous et l'adresse là où on veut arriver. Lorsqu'on cherche l'adresse IP à travers tous les routeurs, on veut savoir où est cette adresse IP, qui est l'opérateur. Alors on fait le mapping inverse. On prend adresse IP et on trouve le nom correspondant.

Imaginez un tableau de host. Si on a un fichier avec une liste de noms et des adresses IP, on veut faire un mapping vers l'avant, c'est relativement facile. On a un nom, on fait le mapping avec l'adresse IP, on cherche dans les fichiers jusqu'à ce qu'on trouve le nom et voilà, on arrive à l'adresse IP. Mais que se passe-t-il si on veut faire le mapping à l'inverse ? C'est facile aussi. On cherche l'IP et à côté, on a leur nom. Alors ceci fonctionne très bien avec le tableau du host.

Que fait-on avec le DNS ? Je vais revenir en arrière pour voir l'exemple de l'espace de noms. Comment est-il structuré, cet espace ? D'une manière faisant en sorte que les noms de domaine soient faciles à trouver. Si je veux chercher www.example.com, je vais à la racine et je descends jusqu'à ce que j'arrive. Mais que se passe-t-il avec une adresse IP ? Qu'est-ce que je fais ? La réponse, c'est que ceci, on ne peut pas le faire avec le DNS de la manière dont je vous ai montré parce qu'on ne peut pas chercher une adresse IP. Alors il fallait trouver la manière de transformer les adresses IP comme des noms de

domaine pour les chercher comme des noms de domaine. Et c'est exactement ce que nous avons.

Alors on a d'autres types de registres qui s'appellent PTR. Les adresses IP vont dans des noms de domaine spéciaux avec des registres PTR qui nous permettent de chercher des adresses IP comme des noms de domaine et trouver le nom correspondant. Alors les adresses IPv4 vont avec un nom de domaine qui s'appelle in-adder, et IPv6 a une autre structure.

Je vais vous montrer un exemple. Ici, vous avez l'arbre de l'espace de noms et je vous montre une partie toute neuve que vous ne connaissez peut-être pas. À droite, on a example.com que nous connaissons déjà. Mais après, il y a le domaine inadder.arpa et là, on aurait le registre PTR d'example.com. Alors à droite, on a un registre qui nous dit : « L'adresse IP est à 192.0.2.7. » Alors on sait quelle est l'adresse.

Mais que se passe-t-il si je veux savoir quel est le nom du domaine qui correspond à cette adresse 192.0.2.7 ? Alors je dois transformer cette adresse IP dans un nom de domaine. Qu'est-ce que je fais ? Je prends cette adresse IP et je la retourne en arrière et j'ajoute le in-adder.arpa et je cherche mon fichier PTR. Alors tout le monde comprend ce que je viens de vous décrire. Mais si j'ai un espace de nom attribué à une personne, les RIR

coopèrent pour administrer ce domaine. Et nous pouvons ainsi avoir une zone qui correspond à une adresse IP déléguée.

Par exemple, on m'a attribué 192.0.2/24, tout ce qui commence avec 192.0.2. Donc ce cas-là, j'aurai ce domaine d'arpa 192.0.2 attribué à moi-même. Et si je veux faire le mapping inverse, il va se passer la même chose. C'est un peu bizarre mais cela marche. La plupart des gens peuvent penser que le mapping inverse n'est pas si important que le mapping vers l'avant. Le mapping vers l'avant nous permet de taper un nom de domaine sur un site web et arriver dans le site web. Et le mapping inverse est fait pour détecter des problèmes. Normalement, les gens ne le font pas ; ce sont les ingénieurs de système qui font le mapping inverse.

Il y a d'autres types de registres, par exemple je vais vous montrer quels sont les autres types de données que l'on a introduites dans le DNS. Ici, nous avons un exemple de fichier de zone pour une zone très petite. C'est un fichier de zone pour la zone example.com. Et je n'ai pas parlé en détail de tous ces registres mais c'est une toute petite zone semblable à toutes les zones sur internet parce que la plupart des noms de domaine sur internet, et probablement il faut faire deux choses. On a un serveur de sites web et l'idée, c'est de recevoir des courriels. Il y a des domaines qui font bien d'autres choses aussi, qui ont beaucoup de noms mais la plupart des noms de domaine, par

exemple mon nom de domaine personnel que j'utilise pour le courrier électronique, et bien je reçois mon courrier, j'ai un tout petit site web et cela suffit. Alors ma zone personnelle est semblable à celle que nous voyons ici. C'est la seule chose dont nous avons besoin à la maison.

On a une adresse IP example.com là où il y le serveur web. Alors vous pouvez voir le fichier de zone et vous pouvez supposer que 192.0.2.7 est l'adresse IP du site web et puis vous avez des registres supplémentaires qui disent où est-ce qu'il faut envoyer le courrier électronique qui doit arriver à example.com.

Maintenant, je veux vous parler du processus de résolution. Et c'est la manière de chercher des choses dans le DNS dans l'exemple du DNS dont je vous ai parlé au début de la séance, les résolveurs et les résolveurs stub, les résolveurs récursifs et les résolveurs faisant autorité.

Ce qui est important, c'est qu'une requête DNS compte sur trois paramètres : un nom de domaine comme example.com pour déterminer le type de données que nous recherchons ; il y a aussi la classe dont je n'ai pas parlé. C'est une manière qu'on pourrait utiliser pour aller avec le DNS sur d'autres réseaux qui n'ont pas été utilisés. Cela existe dans le DNS.

Donc dans ce cas particulier classe, cela va toujours être classe internet et il ne faut pas s'en soucier. C'est tout simplement le

nom de domaine et le type de données qui sont importants. Si nous faisons une requête au DNS, il faut toujours spécifier le nom de domaine et le type.

Les résolveurs stub, c'est notre téléphone, la machine, le réfrigérateur, tout se qui se connecte sur l'internet et qui doit transformer des noms en des adresses et d'autres types d'informations. Tous ces types de dispositifs ont des résolveurs stub et ils envoient des requêtes récursives pour que les résolveurs récursif écoutent. « Il faut que j'aie une réponse ou une réponse d'erreur. Je ne peux rien faire avec une réponse partielle, j'ai besoin d'une réponse complète. »

Les résolveurs récursifs sont plus intelligents et ils peuvent accepter des réponses partielles appelées des dérivations. Ils envoient une requête qui indique qu'ils sont disposés à accepter des réponses partielles des dérivations. Comme je l'ai dit tout à l'heure, si nous cherchons quelque chose sur le DNS, nous commençons par la zone racine et continuons à avancer. Nous suivons la délégation par les pointeurs de délégation.

Il y a des serveurs faisant autorité pour la zone racine qui ont toute l'information de la zone racine. C'est ce que signifie faisant autorité ; c'est les serveurs de la zone racine. Lorsque nous commençons la résolution dans la zone racine, il faut que nous puissions nous connecter sur un serveur de noms racine. Et

comment faisons-nous pour trouver ces serveurs de zone racine ? La réponse est qu'il faut les configurer. Il n'y a pas moyen de les découvrir, il faut les configurer dans tous les serveurs de noms récursifs. Ceci est différent des autres paramètres du réseau.

Par exemple quand mon téléphone s'est branché sur le réseau Wi-Fi de l'ICANN, il y a eu tout un protocole spécial qui a été utilisé de configuration [inintelligible]. Et mon téléphone a dit au réseau : « Je suis un nouveau dispositif sur ce réseau. J'ai besoin d'une adresse IP. » Et le réseau lui a dit : « Bon, très bien. Voilà votre adresse IP et voilà les paramètres de configuration que vous devez connaître. », ce qui inclus l'adresse IP du serveur de noms récursif. Donc de cette façon, un dispositif ignore tout et le réseau lui fournit toute l'information nécessaire. Ce n'est pas le cas pour les serveurs de noms récursifs. Nous ne pouvons pas activer un réseau de ce type sans la configuration. Il faut savoir quels sont les serveurs de noms racine et quelles sont les adresses IP. Alors il y a un fichier spécial dont tous les serveurs de noms récursifs ont besoin qui ont les adresses et les noms des serveurs de noms racine.

Si vous installez un serveur récursif sur un équipement Lenox, quelqu'un a déjà fait tout cela pour les logiciels de tous les serveurs avec toute l'information de la zone racine. Quelques

résolveurs récursifs ont les adresses IP et les noms du serveur racine. Mais c'est le URL où vous pouvez chercher le nom.

Voilà comment cela se voit. Il y a 13 serveurs de noms racine. Il s'agit de 13 serveurs faisant autorité pour la zone racine. À gauche, nous avons la zone racine et à droite, nous avons 13 registres DNS qui sont les noms des serveurs qui s'appellent a.root-servers.net etc., et cela continue. Et puis vous pouvez mettre en dessous les adresses IPv4 et les adresses IPv6. Alors tous les serveurs de noms récursifs ont une adresse IPv4 et une adresse IPv6. Les résolveurs récursifs, pour pouvoir faire une résolution, ont besoin de savoir les noms des serveurs de la zone racine.

Nous allons faire un petit détour pour parler de la zone racine et de la manière dont l'information parvient à la zone racine. Qu'avons-nous là ? Nous avons l'information concernant les zones des domaines de premier niveau. Nous avons des registres DNS pour les TLD. Il est complexe de gérer la zone racine. Il y a ici deux institutions, l'ICANN qui est l'opération des fonctions de l'IANA – et la PTI joue ce rôle – et Verisign sont les responsables de la maintenance de la zone racine.

Verisign était une entreprise qui avait un autre nom qui s'appelait Network Solutions et qui a été acheté par Verisign à l'an 2000. Le contrôle des fonctions de l'IANA était mené à bien

par l'Université de la Californie du Sud avant la création de l'ICANN. Donc c'est un accord très ancien et cette manière de travailler est un peu complexe mais c'est la manière dont la zone racine fonctionne.

Donc ces deux organisations, l'ICANN et Verisign, coopèrent pour mettre les données dans la zone racine pour créer le fichier de la zone racine.

Puis nous avons besoin des serveurs faisant autorité et il y a 12 organisations qui exploitent ces serveurs faisant autorité. Il y a une organisation qui exploite tous les serveurs faisant autorité.

Prenons par exemple .com. Moi, j'ai travaillé chez Verisign donc je sais comment cela fonctionne. Verisign exploite tous les serveurs faisant autorité de .com. Il y a une autre zone comme... Par exemple, de nombreuses entreprises font ce qui suit. Elles exploitent certains serveurs faisant autorité par eux-mêmes ou ils entiercent cette autorité ou il y a des tierces parties multiples pour avoir une redondance. Mais il y n'y a pas 12 organisations. C'est assez peu courant.

Nous avons ici les lettres des 13 serveurs de la zone racine qui s'appelle a.root-servers.net, on l'appelle aroot comme abréviation. Et il s'agit d'organisations qui exploitent ces serveurs. C'est un groupe intéressant d'organisations qui n'ont rien en commun, sauf l'exploitation des serveurs des zones

racine. Vous pouvez avoir des entreprises commerciales, des institutions éducatives, des organisations à but non lucratif, des ISP, certains départements du gouvernement des États-Unis. Il y a enfin... C'est varié. Et encore une fois, cela date d'il y a de longues années, une vingtaine d'années.

Cette liste d'opérateurs n'a pas eu de changement depuis 20 ans. Il y a toute une série de thèmes complexes qui concerne tout cela que je ne peux pas vous expliquer dans le détail parce que nous n'avons pas suffisamment de temps. Nous avons 13 serveurs racine et 12 exploitants ou 12 opérateurs parce que Verisign travaille avec le groupe A et le groupe J.

Voilà les serveurs racine et les opérateurs de ces serveurs racine. Si vous voulez avoir davantage d'informations, vous pouvez rentrer sur root-servers.org et vous allez trouver toute cette information.

Je veux vous montrer à un niveau très général comment on fait des modifications dans la zone racine. Toute l'information de la zone racine se rapporte aux TLD. Donc si un TLD veut faire une modification, veut ajouter un serveur faisant autorité pour son TLD ou éliminer un serveur faisant autorité ou changer le nom d'une adresse IP d'un serveur, il change l'opérateur de fonctions de l'IANA qui est la PTI, c'est-à-dire ce que gère l'ICANN. Et la PTI fait plusieurs vérifications et fait des contrôles et elle met à jour

la zone racine et elle envoie cette information au responsable de la gestion, et c'est Verisign. Verisign fait d'autres mises à jour, il met à jour ce fichier de la zone racine et il diffuse cela. Et après, les 13 serveurs racine téléchargent ce fichier et l'exposent. C'est une description très générale du processus. Je vous l'explique pour que vous ayez une idée de la manière dont les organisations coopèrent pour que cela fonctionne. Mais le processus est beaucoup plus complexe que ce que nous pouvons voir ici. C'est un petit détour que nous avons pris pour aller vers la zone racine.

Maintenant, parlons de la résolution. Supposons que nous avons un téléphone mobile, un téléphone portable, en bas à gauche, et quelqu'un a ouvert le navigateur de son portable et écrit www.example.com. Alors le navigateur web appelle le résolveur stub qui est un code simple. Nous voyons qu'il s'agit ici d'un appel API, une phase de programmation d'application. C'est un petit programme qui s'appelle résolveur stub. C'est une petite fonction. J'ai besoin par exemple de l'adresse www.example.com.

Donc le résolveur stub est très simple. Il ne connaît que l'adresse IP de plusieurs serveurs récursifs auxquels il doit envoyer sa requête. Donc ce résolveur stub transforme cela en une requête et il envoie cette requête au DNS, au serveur de noms récursif qui est, pour ce cas particulier, 4.2.2.2. C'est l'adresse IP d'un

résolveur de noms récursif. Et c'est un résolveur ouvert vers lequel n'importe qui peut envoyer une requête.

Donc le résolveur stub demande au résolveur récursif de donner une réponse à notre consultation. Pour que l'exemple soit plus intéressant, rappelez-vous que les résolveurs récursifs ont une mémoire cachée. Mais supposons que ce résolveur récursif vient d'être mis en fonction, il n'a rien dans sa mémoire cachée. Il ne connaît que les adresses IP et les noms. Donc il prend l'un des serveurs de la zone racine, il choisit celui-ci par exemple et il lui adresse la même requête qu'il vient de recevoir du résolveur stub.

Donc le serveur racine ne peut pas répondre à cette requête de manière directe parce qu'il ne connaît pas l'adresse de www.example.com. Mais il sait quelque chose à propos de .com parce que la zone racine possède la délégation de .com. Alors le serveur racine peut faire un détour pour aller à .com. Il peut dire : « Voilà les serveurs de noms et les adresses IP des .com. » Il met cela dans la mémoire cachée pour pouvoir utiliser cette information dans l'avenir et il fait ensuite ce que nous appelons un suivi de la dérivation. Il choisit un serveur .com et il lui envoie la même requête. Il l'envoie à ce serveur qui s'appelle c.gtld-servers.net. C'est comme cela qu'il appelle les noms des serveurs. Ils utilisent la même forme de noms des serveurs racine.

Voilà donc les noms des serveurs .com. Alors le résolveur récursif a choisi un serveur .com et il lui a envoie la même question qu'il a reçue du résolveur stub. Et il lui demande l'adresse IP de www.example.com. Celui-là, il ne connaît pas non plus l'adresse IP d'example.com mais il sait quel est le serveur de noms faisant autorité et lui envoie cette liste. Il envoie la liste pour que le résolveur récursif puisse aller dans sa mémoire cachée et il continue à faire cette dérivation qui envoie cette requête pour la troisième fois à un résolveur faisant autorité qui est NS1.example.com.

Maintenant, ce serveur connaît l'adresse IP de www.example.com et il donne l'exemple au résolveur récursif qui l'a saisi dans sa mémoire cachée, qui le donne à l'application et l'application connaît l'adresse IP du serveur web. Elle peut contacter et télécharger une page web et ainsi de suite. Voilà le processus de résolution.

Voilà, c'est une version simple, c'est un exemple simple. Comme la plupart des choses que l'on fait sur le DNS, cela peut être très complexe. Mais ici, ce qui est important, c'est que nous sommes partis de la racine et que nous allons vers le bas. Mais on n'est pas toujours obligé de commencer par la racine parce que nous nous servons de la mémoire cachée.

J'ai mentionné tous les points auxquels on faisait mémoire caché. Mais si quelques vient par exemple à ftp.example.com, le résolveur stub prépare sa requête au DNS et l'envoie au résolveur récursif. Mais il se rappelle tout ce que le résolveur récursif a. Il sait quels sont les serveurs .example.com. Donc il va directement à NS1.example.com et il va directement au serveur qui possède cela, reçoit la réponse et il l'envoie au résolveur stub qui l'envoie à l'application. Mais le fait d'avoir de l'information dans la mémoire cachée accélère les processus. S'il n'y avait pas une mémoire cachée, tout serait beaucoup plus lent sur l'internet.

Et ma dernière diapositive donne une description générale de ce que nous avons dit. Nous avons parlé du DNS mais si nous parlons des noms des domaines, nous pensons dans le contexte de l'ICANN, nous réfléchissons et nous pensons à quelque chose d'autre que les serveurs faisant autorité. Nous réfléchissons à tout le panorama qui inclut les registres, les bureaux d'enregistrement, les titulaires de noms de domaine. Et ces titulaires de noms de domaine connectent les bureaux d'enregistrement avec les sites web pour acheter des noms de domaine. Et les bureaux d'enregistrement communiquent avec le registre. Et le registre une base de données où sont enregistrés les noms de domaine et l'information pertinente du registre. Le registre possède cette information dans la base de

données par l'intermédiaire d'un serveur de noms faisant autorité. Et c'est ce qu'utilisent les résolveurs récursifs pour faire leurs requêtes.

J'espère que cela vous aura un peu expliqué comment les choses dont nous avons parlé sont insérées dans le tableau général. C'est la fin de mes diapositives. Nous avons encore le temps. Je serai très heureux de répondre à vos questions.

CATHY PETERSEN : Avant de poser une question, je vous prie de dire votre nom et à quelle organisation vous appartenez.

MATT LARSON : Vous avez faim, vous tous ?

MALISA RICHARDS : Moi, j'ai une question. Je m'appelle Malisa Richards, je suis boursière du Ghana. Quels sont les critères utilisés pour installer un serveur de la zone racine ?

MATT LARSON : Cela dépend de l'opérateur qui chacun ont leur propre politique. Tous les serveurs racines sont Anycast. Il n'y a pas un seul serveur pour une adresse IP d'un serveur racine. Il y a de multiples serveurs. Cette technique Anycast utilise le système de

routage de l'internet qui permet d'avoir plusieurs serveurs avec la même IP qui répondent sur différents points du réseau. Alors tous les opérateurs utilisent des techniques Anycast pour avoir des instances multiples de leur serveur racine.

Et les opérateurs ont des politiques différentes. Je sais que pour le cas de l'ICANN, la racine, elle pourrait... enfin n'importe qui pourrait exploiter une instance de racine L. C'est-à-dire vous achetez le serveur, vous l'installez sur le réseau et ensuite, il faut fournir un espace, une connectivité, une puissance. Puis nous, nous nous occupons de l'exploiter. Il y a différents opérateurs de racine qui ont différentes politiques.

MALISA RICHARDS : Encore une question. Moi je regarde le site web que vous avez recommandé et je vois qu'aux Caraïbes en particulier, il n'y a pas beaucoup de serveurs racine par comparaison à l'Amérique du Sud, l'Amérique du Nord. Pourriez-vous m'expliquer la raison de ce fait ?

MATT LARSON : Je ne sais pas ce qui se passe avec les autres opérateurs. Il se pourrait que les opérateurs de ces réseaux n'aient pas parlé avec les opérateurs de racine pour qu'on installe des serveurs

racine. Dans le cas de l'ICANN, si vous êtes disposé à acheter un serveur, nous pouvons l'installer sur votre base.

Merci beaucoup à vous tous. Est-ce qu'il y a d'autres questions ?

NORMAN WARPUT : Je m'appelle Norman, je viens du Vanuatu. Je suis boursier de l'ICANN. Je vous remercie de la présentation. J'ai une question à propos du processus de modification de la zone racine. Est-ce qu'elle est utilisée aussi pour demander des domaines de premier niveau ?

MATT LARSON : Je n'ai pas très bien compris votre question.

NORMAN WARPUT : Le processus pour la modification des zones racine s'applique aussi aux domaines de premier niveau aux codes de pays ?

MATT LARSON : Du point de vue du DNS, un TLD, c'est un TLD. Nous avons des catégories de TLD comme les ccTLD, les gTLD. Et dans la catégorie gTLD, nous avons ceux qui sont parrainés et ceux qui ne sont pas parrainés. Il y a différentes catégories au point de vue du DNS. De toute façon, un TLD est un TLD, qu'il s'agisse d'un ccTLD ou d'un gTLD ou quoi que ce soit.

Est-ce qu'il y a d'autres questions ?

ANDREW FRASER : Je m'appelle Andrew et je voudrais savoir où est décrit comment on vise un serveur récursif. On le fait par l'intermédiaire du fournisseur d'accès à internet parce qu'il y a différentes alternatives pour les résolveur récursifs.

MATT LARSON : En dernière instance, cela est configuré ou paramétré dans chaque dispositif. Quand le dispositif se connecte sur le réseau, c'est le réseau qui doit nous dire nous. Tous les réseaux vont nous dire, en plus de l'adresse IP quelles est l'adresse IP du serveur récursif. Donc celui qui exploite un réseau devra avoir un serveur récursif.

ANDREW FRASER : Alors mon téléphone est configuré ou préconfiguré par mon fournisseur ?

MATT LARSON : Quand vous vous connectez sur le réseau, votre fournisseur lui dira : « Voilà votre serveur IP et voilà le serveur récursif à utiliser. » Cependant, si vous vous liez, vous pourriez aller au-dessus de cela et vous pourriez choisir un autre. Vous pourriez

rejeter celui-là et vous pourriez choisir un fournisseur différent avec des résolveurs ouverts. Cela a des noms différents et ce sont les entreprises qui exploitent les serveurs récursifs que tout le monde peut utiliser.

La première qui a fait cela, c'était Open DNS. Je me souviens que lorsqu'ils ont fait cela, ma réaction et celle de bon nombre de personnes était la suivante : « Pourquoi veulent-ils faire cela ? » Les serveurs récursifs, c'est un service central du réseau. Pourquoi prendre quelque chose d'aussi important dont moi je dépends ? Pourquoi est-ce que je vais dépendre d'un serveur qui se trouve en dehors de mon réseau ? Et la réponse d'Open DNS était : « Nous pouvons fournir des services supplémentaires. Vous pouvez nous dire « Je ne veux pas résoudre des noms qui correspondent à parier ou contenu pour des adultes, etc. » Nous pouvons faire des choses telles que nous n'allons pas résoudre des noms des domaine qui ont des logiciels suspects. »

Donc ce que proposait le Open DNS était ridicule au début. Mais après, Cisco a acheté Open DNS et a payé beaucoup de millions de dollars. C'est le dernier qui rit qui rit le mieux. Nous avons Google Public DNS et Google a dit qu'il voulait avoir ceci parce qu'il voulait un serveur récursif très fiable pour faire la validation du DNSSEC. Il s'agit d'ajouter l'authentification cryptographique au DNS. Alors Google croit vraiment au DNSSEC et ils disent : « C'est le service que nous proposons qui est gratuit et

vous serez sûrs d'avoir la protection supplémentaire DNSSEC. » Verisign a aussi un service de type, Quad9 en est un autre. Donc si on veut, on peut changer la configuration, le paramétrage du dispositif et utiliser ces serveurs-là. Merci beaucoup.

Est-ce que vous avez d'autres questions ? Merci beaucoup, donc.

CATHY PETERSEN : Merci à vous tous, merci Matt, merci aux transcripteurs, aux interprètes et à l'équipe technique, un excellent travail. Merci beaucoup encore une fois.

[FIN DE LA TRANSCRIPTION]