

Réseaux

DNS

Nicolas GOUVY, Gilles GRIMAUD, Jérémie DEQUIDT

Le Domain Name System (ou DNS, système de noms de domaine) est un service permettant d'établir une correspondance entre une adresse IP et un nom de domaine et, plus généralement, de trouver une information à partir d'un nom de domaine. DNS est un système hiérarchique distribué. Ainsi, un serveur peut déléguer une partie des noms du domaine à un serveur subalterne. On dit d'un serveur de noms responsable de la mise à jour des correspondances nom / adresse IP des machines de son domaine qu'il est le serveur d'autorité pour le domaine (Authorative Serveur).

Le but de ce TD est de vous donner les outils nécessaires à la réalisation du TP 3.

Exercice 1 : UDP et DNS

Pour contacter un serveur DNS, les requêtes doivent être envoyées en UDP sur le port 53.

Q1. Rappelez ce qu'est UDP : nature, niveau dans la couche OSI, inconvénients et nom des paquets UDP.

Q2. En vous remémorant le TP1 (chat multicast en UDP), donnez le nom des principales classes et méthodes Java pour l'utilisation d'UDP.

Format des messages DNS

Dans cet exercice, nous allons voir comment construire une requête DNS en Java. Cela vous servira en TP à implémenter plus rapidement.

Les messages (requête ou réponse) du protocole DNS sont définis dans le RFC1035. La structure d'un message est la suivante :

En-tête	spécifie le type de message (taille fixe 12 octets)
Question	question posée au serveur de nom
Réponse	réponse à la requête
Autorité	information sur les serveurs d'autorité
Additionnel	informations complémentaires

Détails de l'En-tête

Identifiant	Paramètres
QDcount	Ancount
NScount	ARcount

Chaque champ de l'entête est codé sur 16 bits.

- identifiant est un entier permettant d'identifier la requête.
- paramètres contient les champs suivant :
 - QR (1 bit) : indique si le message est une question (0) ou une réponse (1).

- OPCODE (4 bits) : type de la requête (0000 pour une requête simple).
 - AA (1 bit) : le serveur qui a fourni la réponse a-t'il autorité sur le domaine?
 - TC (1 bit) : indique si le message est tronqué
 - RD (1 bit) : demande d'une requête récursive.
 - RA (1 bit) : indique que le serveur peut faire une demande récursive.
 - UNUSED, AD, CD (1 bit chacun) : non utilisés.
- QDCOUNT : nombre de questions.
 - ANCOUNT, NSCOUNT, ARCOUNT : nombre d'entrées dans les champs "Réponse", "Autorité", "Additionnel".

Détails du champs requête/réponse

Une requête / réponse est représentée de la manière suivante :

- Label. C'est ici qu'on indiquera le nom à résoudre pour le serveur DNS Par exemple www.lifl.fr donne 0377 7777 046c 6966 6c02 6672 en hexadécimal. Les points ne sont pas codés par contre on utilise des séparateurs (02 03 04 . . .). Ainsi le premier octet 03 indique qu'il y a 3 octets avant le nouveau séparateur (i.e. 77 77 77 soit www), de même 04 indique qu'il y a 4 octets avant le nouveau séparateur (6c 69 66 6c soit lifl) et le dernier séparateur (02) indique qu'il y a 2 octets après : 66 72 soit fr.
- 00 pour indiquer la fin du label.
- Type (16 bits) pour indiquer le type de la requête. 0x00 01 (=host address) pour ce TD/TP
- Class (16 bits) indique le type du protocole. 0x00 01 (=internet) pour ce TD/TP

Q3. Donnez le label correspondant à l'adresse `http://fr.yahoo.com`.

Q4. Ecrire votre fonction Java qui permet de donner le Label (sous forme d'un tableau de byte) correspondant à une adresse donnée.

Q5. Ecrire votre fonction Java qui permet d'écrire une requête.

Détails du champs requête/réponse

Les champs Réponse, Autorité, Additionnel sont tous représentés de la même manière :

- Nom (16 bits) : Pour éviter de recopier la totalité du nom, on utilise des offsets. Par exemple si ce champ vaut C0 0C, cela signifie qu'on a un offset (C0) de 12 (0C) octets. C'est-à-dire que le nom en clair se trouve au 12ème octet du message.
- Type (16 bits) : idem que pour le champ Question.
- Class (16 bits) : idem que pour le champ Question.
- TTL (32 bits) : durée de vie de l'entrée.
- RDLenght (16 bits): nombre d'octets de la zone RDData.
- RDData (RDLenght octets) : réponse

Exercice 2 : Réponse DNS

Le serveur va répondre à votre requête par un paquet UDP retourné sur le même port que celui d'émission.

Q1. Ecrire votre fonction Java qui permet d'afficher à l'écran l'adresse ip correspondant à notre requête. Vous pouvez maintenant utiliser le flu `ByteArrayInputStream` pour vous aider.

Q2. Réalisez un programme qui prend un nom de domaine en paramètre et retourne l'adresse ip après avoir fait une requête/ une analyse de la réponse DNS.