

Service des réseaux

Serveur DNS

L'app client fait le paramétrage de 4 éléments fondamentaux

- l'adresse IP
- l'adresse masque
- le gateway par défaut
- le serveur DNS

Alors pourquoi ce serveur DNS existe ?

Soit la topologie :

Une machine contenant l'application client, une machine contenant l'application serveur

Et un réseau qui connecte entre ces deux.

Alors le client essaye de se connecter au serveur il veut se connecter à
http// IPserveur :8080(port de ce site web)

Client doit savoir 3 paramètres afin qu'il peut communiquer avec le serveur

- adresse ip du serveur(IPserveur)
- numéro du port (8080)
- Protocole (http)

Remarque :

Les protocoles de communication possible sont http/https/FTP/FTPs

Notre objectif est de simplifier cette @URL aux maximum pour faciliter la communication

⇒ Première simplification : on associe à chaque protocole de communication un port par défaut

Pour un serveur web http on a le port par défaut 80

URL : http//IPserveur :80

Mais on note que le port par défaut est connu partout dans le monde ce qu'il représente le protocole http

⇒ Deuxième simplification : on élimine 80 (ainsi on ne peut pas éliminer le protocole)

Cela nous donne URL : http// IPS

Alors le client doit savoir 2 paramètres pour communiquer avec le serveur :

-adresse ip du serveur

-protocole de communication utilisé

*Pour l'ip serveur

- On peut le garder tel qu'il est
- Mais ça sera plus facile à l'utilisateur si la machine était identifiée par un nom significatif au lieu d'une adresse IP

⇒ Troisième simplification

URL : http: // Nom.Serveur

Nom.serveur représente une @IP

Il faut faire la correspondance @IP et nom

On l'appelle : La résolution

Une fois la résolution est faite – la communication client-serveur se produit

Alors comment se produit la résolution ?

Il faut avoir un composant qui fait la correspondance entre l'adresse IP et le nom

Et pas intégrée dans une machine/app spécifique mais plutôt doit être un service résea (cad client et serveur)

Alors le processus se transforme en :

Une application client donne le nom du domaine à la place de l'adresse ip du serveur à une machine appelé DNS résolveur qui fait la résolution précédemment mentionnée.

Ce résolveur est une composante logicielle du système d'exploitation qui a pour rôle la résolution des noms en contactant un serveur DNS

Et pour URL http//serveur1

Résolveur contacte un serveur DNS : « donne moi une adresse IP qui correspond au serveur 1 »

La résolution est stockée dans la mémoire cache (temporaire) du résolveur, ainsi si une autre application client demande la résolution du même nom elle est prête

Notons bien que la résolution prend beaucoup du temps

Et enfin le serveur DNS va recevoir des requêtes du résolveur afin de trouver réellement la correspondance nom et adresse IP

⇒ Résolveur intermédiaire entre client et serveur DNS

Pourquoi ne pas avoir communication directe entre client et serveur ?

Supposons qu'on a 2 applicationn locales dans la même machine veulent accéder au même nom dans 2 instants différents alors la résolution est de 2 fois mais la résolution prend beaucoup de temps

*Cependant avec le résolveur :

- la résolution se produit une seule fois

- Résultat stocké dans la mémoire cache peut être exploité plusieurs fois

*on utilise le résolveur afin de

- minimiser la charge sur le serveur DNS

- optimiser la communication client<->serveur

- simplifier le code de l'application client

Soit le Protocole DNS implémenté entre Resolveur et serveur DNS

Le Résolveur doit savoir :

- le num Port du serveur DNS

- l'adresse IP du serveur DNS

Le client de l'autre part configuré à l'aide du DHCP ou manuellement a les paramètres suivantes :

- ipClient/masque

-table de routage (default gateway)

- adresse IP du serveur DNS

Lors de la configuration du client à l'aide du DHCP le client enfin a trouvé son adresse IP il reçoit alors son DHCP acknowledge ici il reçoit les champs suivante :

-adresse ip client

-masque

-default gateway

-adresse ip du serveur DNS

Ainsi le résolveur sait que : quand il a besoin de faire une autre résolution il va contacter ip DNS

Communication entre résolveur et serveur DNS via UDP

Le résolveur reçoit les requêtes du serveur DNS (contenant la correspondance nom<-> adresse IP)

Résolveur doit renvoyer une réponse au serveur DNS

Ce serveur DNS doit avoir la table de correspondance nom-adresse IP

Mais avec un Internet

Ce n'est pas possible d'avoir la base de donnée de toutes les résolutions dans une seule machine

➔ Il faut diviser la Base de donnée à des partie des serveurs

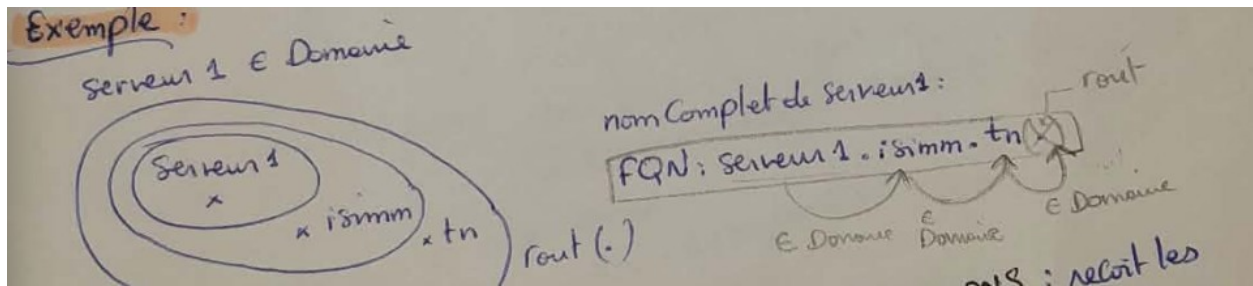
Pour cela on a suici l'exemple des fichiers et répertoires pour accéder à un nom précis

La machine appartient à un domaine qui peut être un sous domaine d'un domaine supérieur

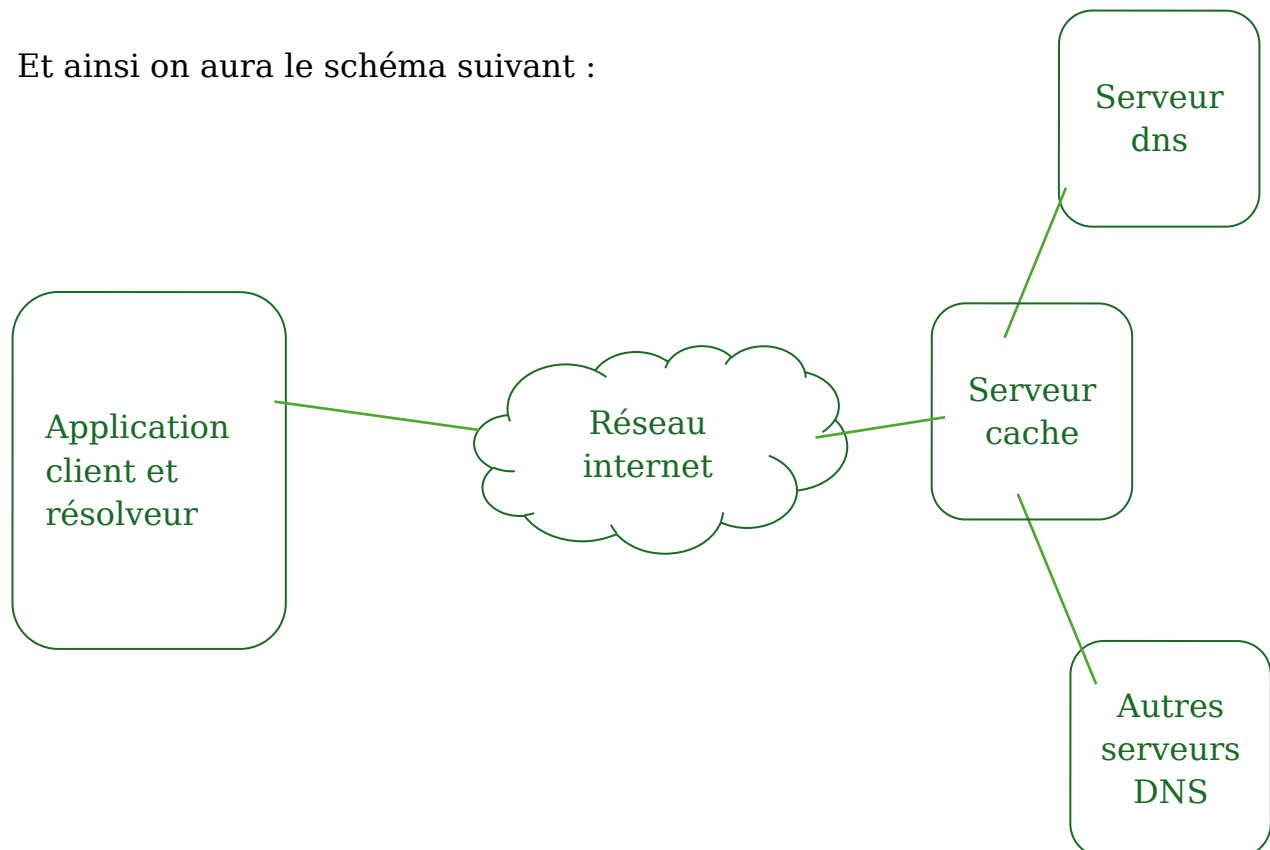
Le nom

On associe à chaque domaine un serveur DNS et enfin un domaine racine qui regroupe tous les sous domaines.

Exemple :



Et ainsi on aura le schéma suivant :



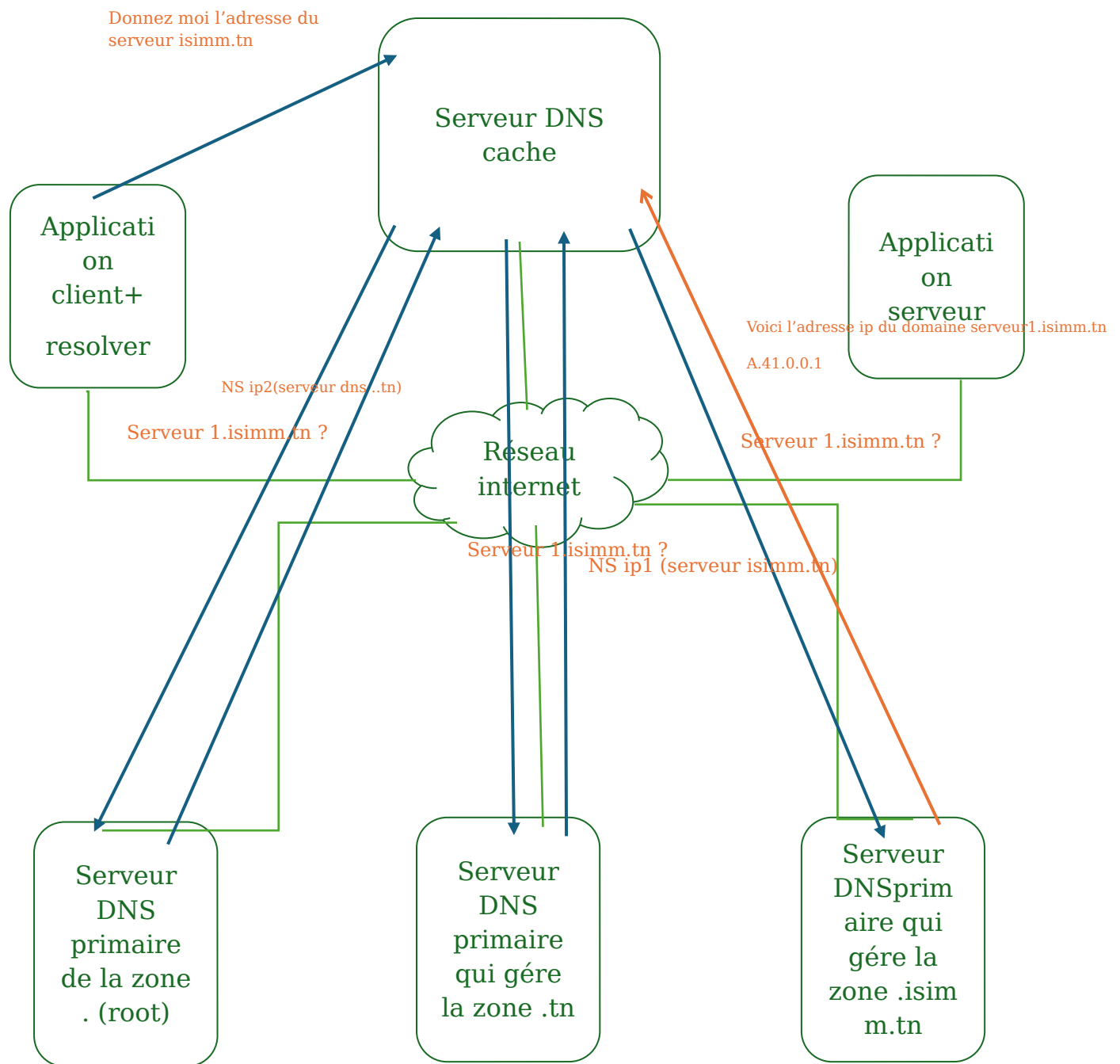
Le serveur DNS reçoit les requêtes du résolveur

Le Serveur DNS cache qui fait la résolution ; intermédiaire entre résolveur et serveur DNS primaire

Les autres serveurs DNS reçoivent les requêtes de serveur DNS contiennent une base de donnée qui contient une table de correspondance entre le nom du domaine et l'adresse IP serveur

Mais comment le serveur DNS cache peut savoir que la zone du site web par exemple est gérée par ip 1 ? pour avoir la correspondance d'adresse ip au serveur1

Il ya un serveur DNS root (racine) qui gère le domaine root
Soit ce sh2 montrant comment trouver le site web isimm.tn



Explication :

Ici le serveur DNS cache ira au serveur DNS de la zone racine lui demande l'adresse du serveur1.isimm.tn

Il lui répond « non mais je connais l'adresse ip2 du serveur qui gère .tn » alors délégation d'ip zone à .tn

Ici on s'adresse au serveur DNS de la zone.tn avec la même question de recherche

.tn répond « je n'ai pas l'adresse mais je connais l'adresse ip1 du serveur responsable de isimm»

Ici on s'adresse au serveur DNS de la zone isimm.tn

Reponse finale : On a trouvé la correspondance de nom serveur 1 avec

L'adresse ip 41.0.0.1 et on l'envoie au serveur DNS cache qui la met dans sa mémoire cache et l'envoie au résolveur qui la met dans sa mémoire cache et l'envoie au résolveur qui la met dans sa mémoire cache à son tour(ainsi si une application demande la résolution du même nom pas besoin de refaire toutes ces étapes une autre fois)

Remarque :

Le type de cette réponse est A

-les réponses NS ne doivent pas être mises dans la mémoire cache. Dans ce cas au moment de l'expiration de cette résolution ip1 vas changer ainsi la zone isimm.tn devient inaccessible pour un certain temps et on ne peut pas faire des résolution d'autres noms appartenant à cette zone

- Les serveurs DNS root ne sont pas gérée par un seul serveur DNS sinon on aura un problème au niveau de la tolérance de pannes

Par exemple si un seul serveur DNS qui gère le root Inthernet si il y a défaillance, panne, problème de réseau toutes les résolution Inthernet des noms arrêtent

Alors la solution :

On peut avoir 13 serveurs DNS 1 primaire et 12 secondaires

-Le serveur DNS secondaire est comme le primaire (gère une zone) mais son fichier zone est située dans le serveur DNS primaire

Cad le serveur DNS secondaire télécharge la base de donnée du fichier zone du primaire

Ainsi si le primaire est non fonctionnel

Alors le secondaire a une copie de la base de donnée du primaire prend la relève du primaire

De même dans le résolveur on a dans sa mémoire cache l'adresse de plusieurs serveurs DNS, s'il n'est pas fonctionnel on contacte l'autre (secondaire)

Il suffit de faire une mise à jour dans le fichier zone du DNS primaire et tous les serveurs DNS secondaires seront mise à jour automatiquement (pas de problème des configurations de synchronisation)