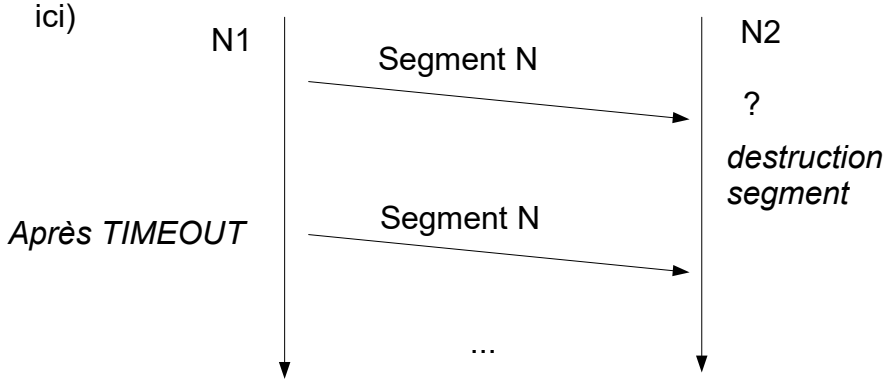
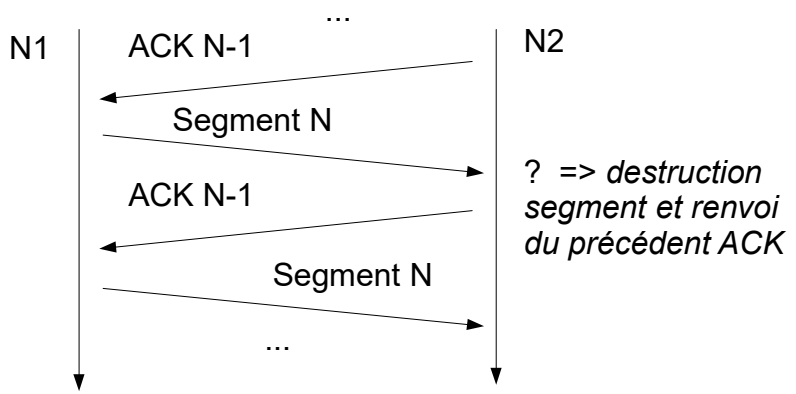
III – TCP

- Si segment reçu avec erreur : comportement identique à non réception avec destruction du segment par le destinataire (N2 ici)



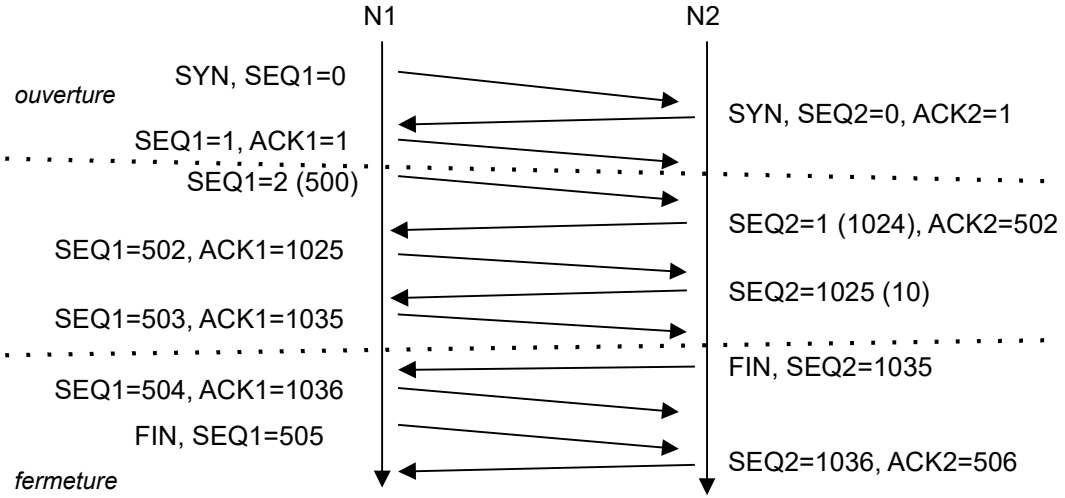
III – TCP

- Avec option NACK (non acquittement), si segment reçu avec erreur : duplication d'ACK. Le destinataire (N2 ici) renvoie le précédent ACK => l'expéditeur (N1 ici) renvoie le même segment aussitôt (sans attente du TIMEOUT)



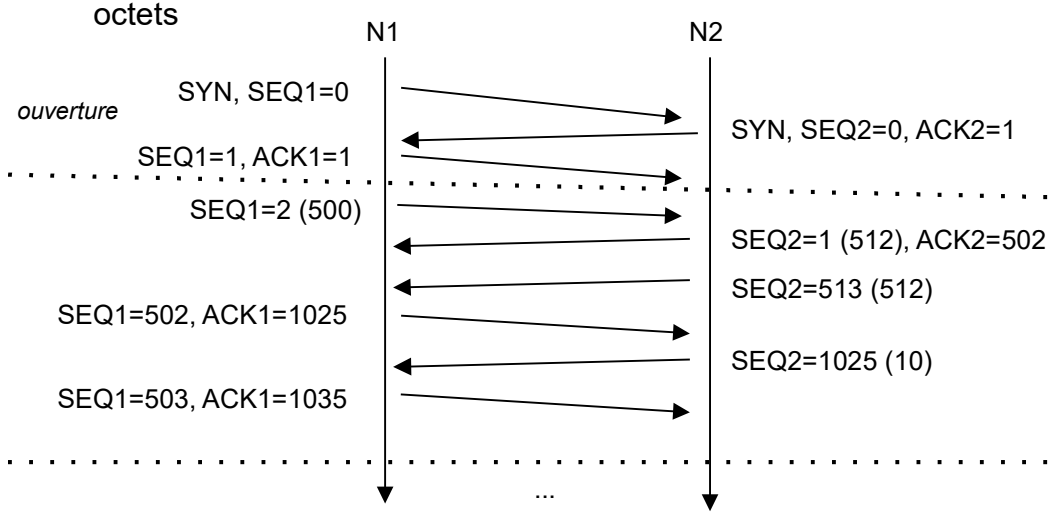
III – TCP - Exemples

- WIN = 1024. Ni option, ni erreur. 1er bloc données de N1 vers N2, 500 octets. 2ème bloc données de N2 vers N1, 1034 octets



III – TCP - Exemples

- WIN = 1024. Option MSS = 512. Pas d'erreur. 1er bloc données de N1 vers N2, 500 octets. 2ème bloc données de N2 vers N1, 1034 octets



III – UDP

- UDP = User Data Protocol (RFC 768)
- UDP offre aux applications réseaux un service sans connexion et sans ACK, donc beaucoup moins fiable que TCP, mais plus rapide que TCP
- Permet des échanges (= datagrammes UDP) bidirectionnels et multipoints

III – UDP

- Format d'un datagramme

en-tête

Port source (16 bits)	Port destination (16 bits)
Longueur totale (16 bits)	Total de contrôle (16 bits)
Données (taille variable)	

III – UDP

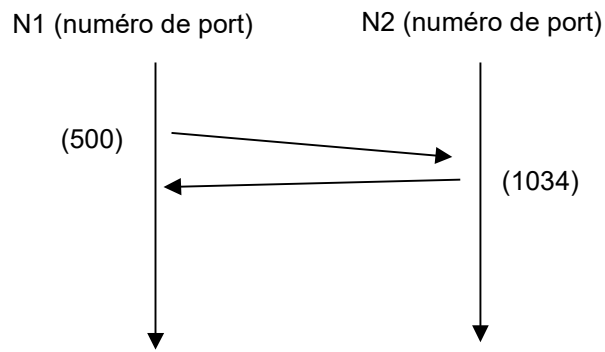
- Échanges
  - Contrairement à TCP, chaque bloc de données issu des applications est intégralement intégré dans un datagramme
  - D'où taille du bloc obligatoirement < 64 kO, sinon bloc pas transmis
  - Pas de connexion à gérer => aucun échange avant ni après l'envoi d'un datagramme (1 seule phase)
- Gestion des problèmes
  - UDP ne gère que le contrôle d'erreur dans le datagramme. En cas d'erreur détectée => uniquement destruction du datagramme. En cas de non réception => aucune action

III – UDP - Exemple

- 1er bloc données de N1 vers N2, 500 octets. 2ème bloc données de N2 vers N1, 1034 octets

Pas de connexion à gérer ; en particulier, pas de préliminaires

Pas de contrôle de flux => ni ACK, ni découpage, etc



IV – DNS

- DNS = Domain Name System (RFC 1034 et 1035)
- Permet l'utilisation de noms à la place d'adresses IP
- Mais noms pas utilisables par les machines du réseau ! => il faut systématiquement les traduire en adresses IP
- Ex. d'utilisation :
  1. on inscrit « www.google.fr » dans barre d'adresse du navigateur
  2. Navigateur fait appel à DNS (au solveur, cf. plus loin)
  3. DNS fournit @ IP de « www.google.fr » au navigateur
  4. Navigateur utilise cette @ IP pour contacter cette machine

IV – DNS

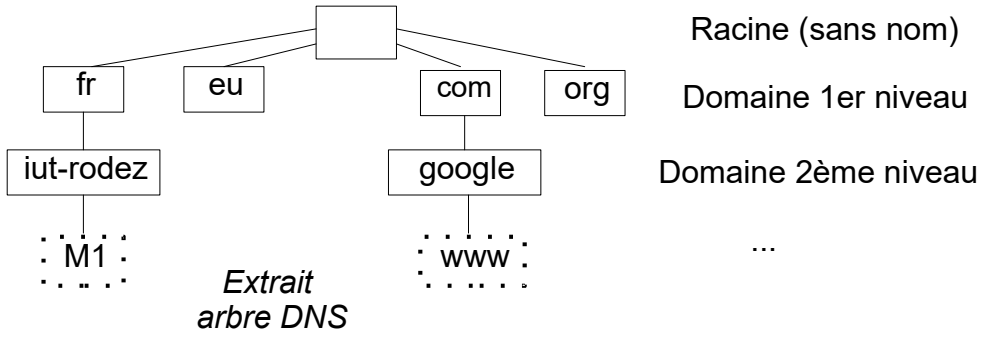
- Remarque « historique »
- Avant DNS, on utilisait aussi des noms. Il fallait inscrire sur chaque machine dans un fichier texte, appelé fichier « HOSTS », les noms et adresses IP de toutes les autres machines
- Ensuite, « HOSTS » est devenu un fichier centralisé, partagé. Mais problèmes car taille énorme !
- De plus, les noms étaient donnés sans concertation ni règle générale
- Ce système avec fichier HOSTS est maintenant inconcevable pour Internet mais est encore utilisé dans les LAN

IV – DNS

- DNS propose :
  - Un système de nommage hiérarchique et arborescent basé sur la notion de domaine, géré par IANA
  - Une BD répartie. De nombreux fichiers, appelés fichiers de zone, disséminés sur Internet, contiennent des noms DNS et leur adresse IP associée (1 nom = 1 adresse IP). Chaque fichier ne contient que des entrées de la même partie de l'arbre DNS. Ces fichiers sont gérés par des serveurs DNS
  - Une application C/S de recherche dans cette BD. Le client est appelé solveur. Il est présent sur toutes les machines « standards ». Le solveur interroge les serveurs DNS pour trouver l'adresse IP correspondant au nom désiré. On parle de résolution du nom

IV – DNS – système de nommage

- Tous les noms appartiennent à un unique arbre. Chaque nœud et feuille représentent un domaine, de niveau hiérarchique « i », identifié par un nom (une étiquette). « i » indique la profondeur dans l'arbre. Chaque feuille représente une interface réseau



Extrait arbre DNS

IV – DNS – système de nommage

- Le domaine de 1er niveau (TLD = Top Level Domain) est :
  - soit générique (gTLD = generic TLD) : edu, gov, com, org, ...
  - soit géographique (ccTLD = country code TLD) : 2 lettres représentant un pays ou un ensemble de pays : fr, eu, de, us, jp, … (norme ISO 3166)
- Un nom complet (= FQDN = Fully Qualified Domain Name) est construit en juxtaposant tous les noms rencontrés en parcourant l'arbre, noms séparés par des points (point = séparateur de niveau dans l'arbre), parcours de la racine jusqu'au nœud considéré :

  nom dom. ième niv. · ... · nom dom. 2ème niv. · nom dom. 1er niv. ·

IV – DNS – système de nommage

nom dom. ième niv. · ... · nom dom. 2ème niv. · nom dom. 1er niv. ·

- En général, le nom le plus à gauche = nom feuille et correspond à une interface réseau (en fait plutôt au nom de la machine)
- Lecture de droite vers gauche, racine vers feuille ou nœud
- Remarquer le dernier point : il sépare le nom de 1er niveau de la racine sans nom ; souvent omis

IV – DNS – système de nommage

En pratique, nom de feuille (dernier niveau) est appelé nom d'hôte (sous-entendu court ou incomplet), tout le reste = suffixe DNS ou nom de zone (différence vue plus loin)

- « Nom d'hôte » prête à confusion. En général, sans plus de précision = nom donné à l'ordinateur (ou plus rarement à une interface réseau). Sinon, on précise « complet » ou « FQDN »
- Bref, en pratique, un nom complet d'une machine =

  nom d'hôte · suffixe DNS (point final omis)

  Ou nom d'hôte · nom de zone (point final omis)

IV – DNS – système de nommage

- Exemple : www.iut-rodez.fr.
- « En théorie » :
  - lecture de droite vers gauche
  - Nom domaine 1er niveau = ccTLD = fr
  - Nom domaine 2ème niveau = iut-rodez
  - Nom domaine 3ème niveau (ici, feuille) = www
- « En pratique » :
  - lecture de gauche vers droite
  - Nom d'hôte (nom feuille) = www
  - Suffixe DNS ou nom de zone = iut-rodez.fr

IV – DNS – BD répartie

- Chaque domaine de 1er niveau est géré indépendamment des autres. Par exemple, l'AFNIC gère le domaine géographique « fr »
- Le lien entre les domaines de 1er niveau est fait par les serveurs racines (13), gérés par IANA et les RIRs.
- Ensuite, chaque domaine gère tout ou partie de son sous-arbre
- Un domaine de niveau « i+1 » ne peut être créé que par le domaine de niveau « i » auquel il est rattaché
- Exemple : « www.iut-rodez.fr. » ; Seul « fr » peut créer « iut-rodez »

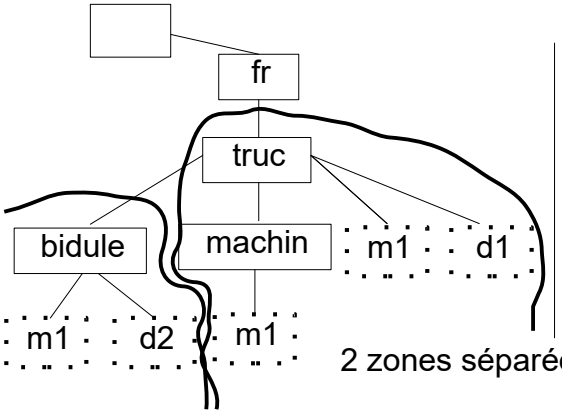
IV – DNS – BD répartie

- En fait, l'arbre DNS n'est pas découpé en niveau mais en zones, en fonction de la présence ou non de serveur DNS et non en fonction des noms de domaine
- Car la BD est gérée par les serveurs DNS
- Une zone correspond à la partie de l'arbre sous contrôle d'un serveur DNS donné. Le nom de la zone correspond au suffixe DNS du nom du serveur
- Une zone peut ne recouvrir qu'une partie d'un sous-arbre

IV – DNS – BD répartie

- Exemple : « d1 » = serveur DNS de zone « truc.fr. »

  « d2 » = serveur DNS de zone « bidule.truc.fr. »



« d1 » gère tous les noms du type « xxx.truc.fr. » (« m1.truc.fr. » ou « m1.machin.truc.fr. », ...)

mais pas ceux du type « xxx.bidule.truc.fr. » (« m1.bidule.truc.fr. », ...) gérés par « d2 »

2 zones séparées

IV – DNS – BD répartie

- Délégation : fait de laisser la gestion d'une partie d'un sous-arbre à un autre serveur DNS => création d'une nouvelle zone et diminution de la zone initiale
- Ex. : « d1 » délègue à « d2 » la gestion de la zone « bidule.truc.fr. »
- Note : créer un sous-domaine n'implique pas obligatoirement une délégation. Ex. de « machin »

IV – DNS – BD répartie

- Un fichier de zone par zone
- Contenu précis d'un fichier de zone : cf TD/TP
- Remarque : un seul serveur DNS est nécessaire pour gérer une zone. Mais par sécurité, on peut en mettre plusieurs. Dans ce cas, un seul fait autorité (autorité donnée par le serveur de niveau au-dessus). Le serveur qui fait autorité est le primaire ou principal ou préféré. Les autres serveurs sont secondaires ou auxiliaires
- Les zones sont indépendantes des réseaux physiques et IP
- Les machines d'une zone peuvent appartenir à des réseaux IP différents

IV – DNS – algo. de recherche

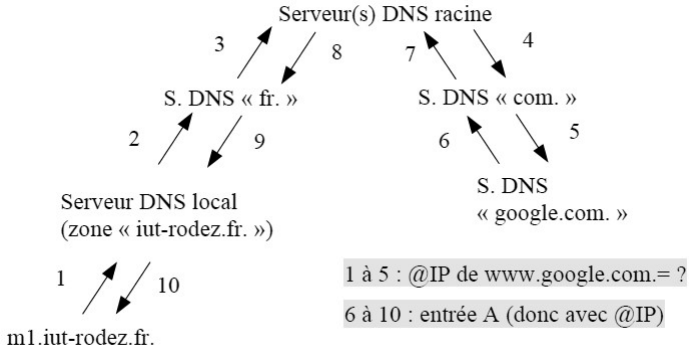
- C'est toujours le solveur qui initie une recherche. Recherche la plus classique : on souhaite connaître l'adresse IP correspondant à un nom DNS = recherche directe. Parfois (beaucoup plus rare), on souhaite connaître le nom DNS correspondant à une adresse IP = recherche inverse
- Algo. de recherche :
  - Le solveur fait appel à son serveur DNS primaire
  - Celui-ci consulte (parcours de l'arbre) plusieurs serveurs DNS afin de trouver celui qui peut avoir la réponse
  - Celui-ci consulte le fichier de zone adapté et retourne l'information recherchée (i.e. l'adresse IP correspondant au nom la plupart du temps)

IV – DNS – algo. de recherche

- Parcours de l'arbre théorique 1 : méthode récursive
- consiste à interroger tour à tour tous les serveurs DNS
- en commençant par le serveur de sa zone (= serveur préféré, principal)
- de manière à remonter l'arbre jusqu'à la racine s'il le faut
- puis à le redescendre jusqu'au « bon » serveur DNS (i.e. celui qui dispose de l'information)

IV – DNS – algo. de recherche

- Parcours de l'arbre théorique 1 : méthode récursive
- Exemple : « m1.iut-rodez.fr. » recherche l'adresse IP de « www.google.com. »

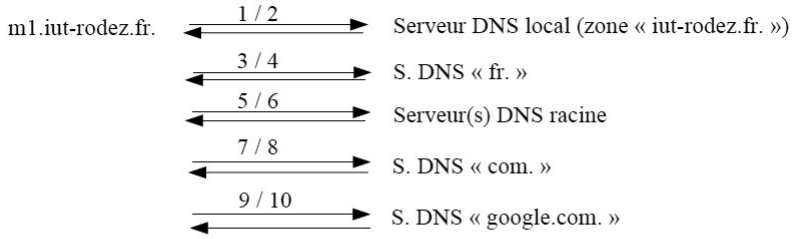


IV – DNS – algo. de recherche

- Parcours de l'arbre théorique 2 : méthode itérative
- oblige la machine demandeuse à gérer elle-même sa recherche
- en commençant par le serveur de sa zone
- chaque serveur DNS se contente de lui donner ses entrées NS ou l'entrée A s'il la connaît
- la machine demandeuse doit faire le tri dans les entrées NS reçues et interroger tour à tour les serveurs DNS qui lui semblent les plus pertinents
- l'arbre est encore une fois parcouru d'abord en le remontant jusqu'à la racine s'il le faut puis en redescendant jusqu'au « bon » serveur DNS

IV – DNS – algo. de recherche

- Parcours de l'arbre théorique 2 : méthode itérative
- Exemple : « m1.iut-rodez.fr. » recherche l'adresse IP de « www.google.com. »



1, 3, 5, 7, 9 : @IP de www.google.com.= ?

2, 4, 6, 8 : entrée(s) NS pour parcourir l'arbre

10 : entrée A (donc avec @IP)

IV – DNS – algo. de recherche

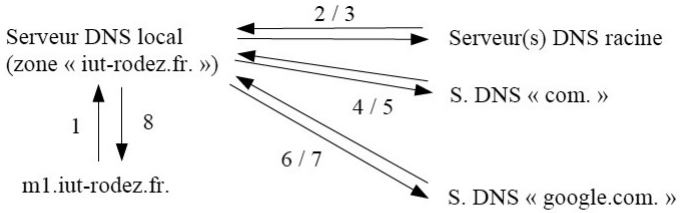
- Comparaison de ces méthodes théoriques de parcours de l'arbre
  - Méthode récursive mobilise trop les serveurs
  - Méthode itérative demande beaucoup à la machine demandeuse
- Inconvénient commun : chaque serveur DNS doit connaître les serveurs DNS en dessous de lui dans l'arbre mais surtout aussi au dessus => problème de gestion et de sécurité

IV – DNS – algo. de recherche

- Parcours de l'arbre utilisé par DNS
- utilise une méthode mélangeant itératif et récursif pour garder le meilleur des 2 :
  - récursif entre le client et le serveur DNS local
- itératif entre ce serveur DNS et les autres serveurs DNS, en commençant directement par la racine (pour éviter l'inconvénient majeur précédemment cité)
- contrepartie de commencer directement à la racine : oblige tous les serveurs DNS à connaître les noms et adresses IP des serveurs racines

IV – DNS – algo. de recherche

- Parcours de l'arbre DNS
- Exemple : « m1.iut-rodez.fr. » recherche l'adresse IP de « www.google.com. »



1, 2, 4, 6 : @IP de www.google.com.= ?

3, 5 : entrée(s) NS pour descendre l'arbre

7, 8 : entrée A (donc avec @IP)

IV – DNS – protocole

- Les messages DNS ont tous la forme suivante :

Identifiant (16 bits)	Drapeaux (16 bits)
Nombre de questions (16 bits)	Nombre de réponses (16 bits)
Nombre de réponses pointant vers un autre serveur (16 bits)	Nombre de réponses autres (16 bits)
Questions (taille variable)	
Réponses (taille variable)	

IV – DNS – protocole

- Dans son fonctionnement de base :
  - Le solveur envoie un message à son serveur DNS primaire avec un nom DNS dans le champ « questions ». « réponses » est vide
  - Le message circule de serveur en serveur. La question n'est pas modifiée. « réponses » contient l'adresse IP du prochain serveur DNS à interroger
  - Lorsque la réponse a été trouvée, le message contient toujours la même question. « réponses » contient l'adresse IP recherchée
- DNS utilise (la plupart du temps) UDP, port 53

IV – DNS – remarques

- En pratique, à chaque adresse IP peut correspondre plusieurs noms DNS. Toutefois un seul est considéré comme l'officiel et est appelé nom canonique. Les autres noms sont appelés alias
- Plus rarement, un nom DNS peut avoir plusieurs adresses IP (cas de Google par ex.). Mais il faut utiliser un type d'entrée particulier, RRSet (RFC 2181), qu'on ne verra pas
- DNS est spécifique à certains réseaux, dont Internet. Il existe d'autres systèmes de nommage et de stockage / recherche de noms. Ex. NetBIOS, fichier LMHOSTS, serveur WINS