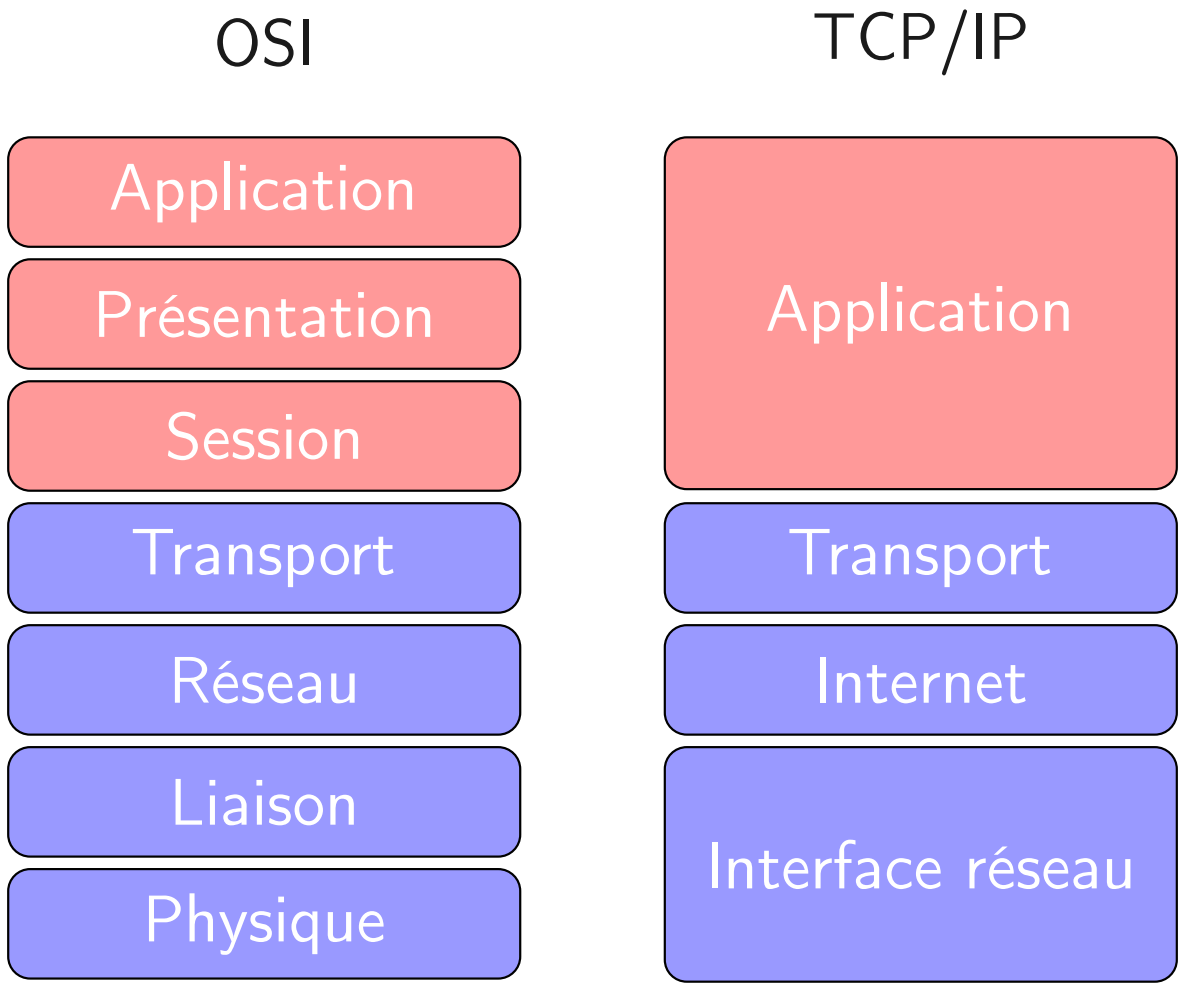
Couche application

Martin Delacourt, Université d'Orléans

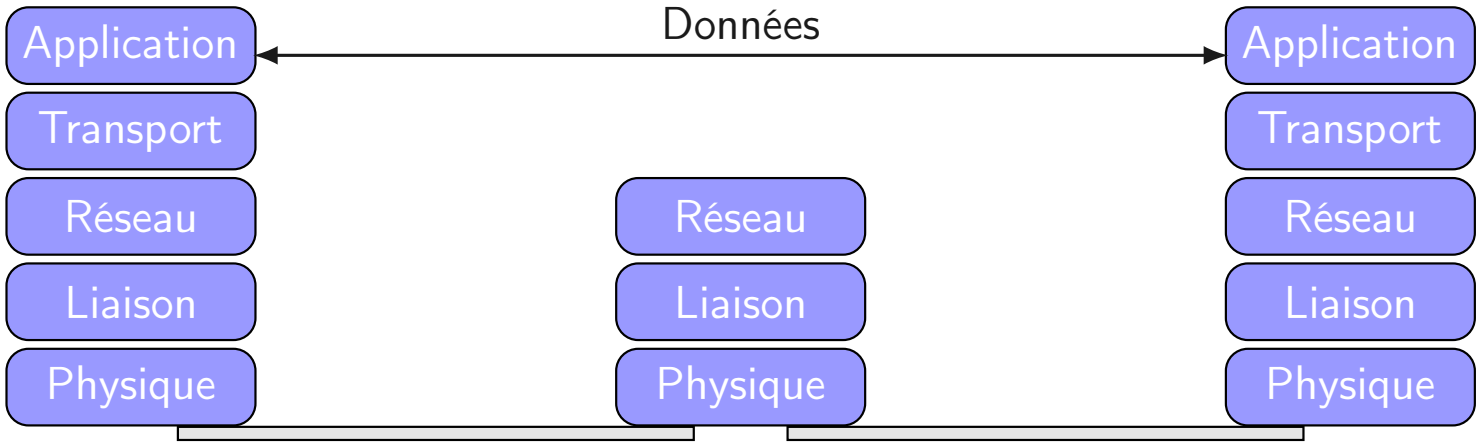
L3 Réseaux — 2023/2024

Return-Path: <mdelacourt@dim.uchile.cl>
Delivered-To: p53252@univ-orleans.fr
Received: from juni.univ-orleans.fr (juni.univ-orleans.fr [194.167.30.176])
by mailper.univ-orleans.fr (Postfix) with ESMTP id F2591936C1
for <martin.delacourt@univ-orleans.fr>; Thu, 30 Aug 2018 15:19:34 +0200 (CEST)
Received: from localhost (localhost [127.0.0.1])
by juni.univ-orleans.fr (Postfix) with ESMTP id 2A5722062F
for <martin.delacourt@univ-orleans.fr>; Thu, 30 Aug 2018 15:19:32 +0200 (CEST)
X-Virus-Scanned: Debian amavisd-new at juni.univ-orleans.fr
Received: from juni.univ-orleans.fr ([127.0.0.1])
by localhost (juni.univ-orleans.fr [127.0.0.1]) (amavisd-new, port 10024)
with ESMTP id 8b8U9FpI23hY for <martin.delacourt@univ-orleans.fr>;
Thu, 30 Aug 2018 15:19:30 +0200 (CEST)
Received: from mxb2-2.relay.renater.fr (mxb2-2.relay.renater.fr [194.214.200.9])
by juni.univ-orleans.fr (Postfix) with ESMTP id 34D712045F
for <martin.delacourt@univ-orleans.fr>; Thu, 30 Aug 2018 15:19:30 +0200 (CEST)
Received: from scfilter.dim.uchile.cl (scfilter.dim.uchile.cl [146.83.7.10])
(using TLSv1.2 with cipher ECDHE-RSA-AES256-GCM-SHA384 (256/256 bits))
(No client certificate requested)
by mxb2-2.relay.renater.fr (a
for <martin.delacourt@univ-orleans.fr>; Thu, 30 Aug 2018 15:19:32 +0200 (CEST)
Received: from localhost (localhost.localdomain [127.0.0.1])
by scfilter.dim.uchile.cl (Postfix) with ESMTP id 421NPV5tdgzyNj
for <martin.delacourt@univ-orleans.fr>; Thu, 30 Aug 2018 10:19:26 -0300 (-03)
X-Amavis-Modified: Mail body modified (using disclaimer) -
scfilter.dim.uchile.cl
X-Virus-Scanned: Scrollout F1 at dim.uchile.cl
Received: from scfilter.dim.uchile.cl ([127.0.0.1])
by localhost (scfilter.dim.uchile.cl [127.0.0.1]) (amavisd-new, port 10024)
with ESMTP id hCruDV75ftrT for <martin.delacourt@univ-orleans.fr>;
Thu, 30 Aug 2018 10:19:13 -0300 (-03)
Received: from gauss.dim.uchile.cl (unknown [10.0.0.22])
(using TLSv1.2 with cipher ECDHE-RSA-AES256-GCM-SHA384 (256/256 bits))
(No client certificate requested)
by scfilter.dim.uchile.cl (Postfix) with ESMTPS id 421NPF5SJPzyVQ
for <martin.delacourt@univ-orleans.fr>; Thu, 30 Aug 2018 10:19:13 -0300 (-03)
Received: from [192.168.80.74] (natlifo.univ-orleans.fr [193.49.83.102])
(using TLSv1.2 with cipher ECDHE-RSA-AES128-GCM-SHA256 (128/128 bits))
(No client certificate requested)
by gauss.dim.uchile.cl (Postfix) with ESMTPSA id 8BB582115B9E
for <martin.delacourt@univ-orleans.fr>; Thu, 30 Aug 2018 10:18:55 -0300 (-03)
DKIM-Signature: v=1; a=rsa-sha256; c=simple/simple; d=dim.uchile.cl;
s=gauss; t=1535635136;
bh=qkaK/xBSLdcrM5/ch5MCiQcCGdvBYn6WctsxoMQr1tY=;
h=To:From:Subject:Date;
b=IpRgOK7sNH6Zw1nKgtkyVE9W11Chw74RZYaiXCjgJdWgvYoANm20H6AarRjMZ7Wji
twehAvkDA/G/Nh3fjCegMk0yXru7flOrF1bc7Ug+47a5nDd3LpPmYNTX5GLgJQYYSB
coRRUHSC5ktX1cqVvcRHpNB4nSIBmhG6VQG6OEpE=
To: Martin <martin.delacourt@univ-orleans.fr>
From: Martin Delacourt <mdelacourt@dim.uchile.cl>
Subject: =?UTF-8?Q?=c2=a1Hola!?=
Message-ID: <d065fa9b-9d9e-b451-b16f-036c5f301dba@dim.uchile.cl>
Date: Thu, 30 Aug 2018 15:18:49 +0200
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101
Thunderbird/52.9.1
MIME-Version: 1.0
Content-Type: text/plain; charset=utf-8; format=flowed
Content-Transfer-Encoding: 7bit
Content-Language: en-US
X-UCHILE-MailScanner-Information: Please contact the ISP for more information

Les modèles


OSI
TCP/IP
Application
Présentation
Session
Transport
Réseau
Liaison
Physique
Application
Transport
Internet
Interface réseau

Couche application (Application)



Les applications que vous connaissez avec leurs protocoles :

- Mail : SMTP
- Web : HTTP
- Chat : XMPP
- Envoi de documents : FTP
- ...

Couche application ?

Interactions entre hôtes dans le but de délivrer un service à l'utilisateur :

- échange de données (fichier, web, mail,...)
- ou d'informations sur le réseau (DNS).
- on suppose que toute la machinerie sous-jacente est fonctionnelle : on sait envoyer un message (du texte, une chaîne d'octets,...) à une adresse IP existante donnée.

Les protocoles de la couche application décrivent ces interactions.

Application réseau

Une application réseau, c'est :

- Un (plusieurs ?) protocole mais aussi...
- des formats, des normes...
- des applications logicielles, en particulier un agent utilisateur (user agent).

Exemple : Le web :

- HTTP
- HTML
- Navigateur web (Opera...), serveur web (Apache...).

Organisation : client-serveur

Le modèle le plus courant est l'architecture client-serveur :

- le serveur est un hôte toujours connecté et disponible ;
- le client se connecte ponctuellement et envoie des requêtes au serveur.
- En général, plusieurs clients, un seul serveur.

Exemples : clients et serveurs web ou mail.

Client-serveur

- Un protocole sur ce modèle décrit les échanges entre client et serveur : synchronisation, taille et contenu des messages...
- Pour être joignable, le serveur doit en général disposer d'une IP fixe.
- Au niveau logiciel : une application serveur, une application client.

Autre modèle : généralisation

- Pour des questions de capacité ou de garantie de service, plusieurs serveurs peuvent se répartir les requêtes.
- Autre paradigme : P2P, les données sont distribuées et les communications ne sont pas centralisées.
- Chaque hôte peut être client ou serveur, voire les deux en parallèle, donc ponctuellement, on peut se ramener au modèle client-serveur.

Design : format

L'application doit spécifier le format des données échangées :

- données textuelles : HTTP, SMTP,...
- XML : XMPP...
- binaire, hexadécimal
- ...

Avec des séparateurs (ou une taille précisée au début du message), des codes de confirmation ou d'erreur...

Design : synchronisation

L'application doit décrire les échanges de messages :

- Mode connecté : échange de questions/réponses dans une connexion établie à l'avance. Protocole TCP de la couche transport pour échanger un flux de données. Exemples : HTTP, SMTP, FTP.

- Mode non connecté : envoi de datagrammes par le protocole UDP de la couche transport. Pas de garantie d'acheminement du datagramme, l'application doit pouvoir fonctionner avec pertes. Exemples : DNS, streaming.

En demandant un service spécifique à la couche transport (débit, sécurité, garantie de transfert), on choisit des contraintes : structure forte ou absence de garantie par exemple.

Sockets

Pour concevoir une application réseau, on utilise des sockets : application logicielle servant d'interface entre le système d'exploitation et le réseau. Une socket est la donnée :

- d'un protocole (UDP ou TCP en général)
- de l'adresse de transport de l'hôte local (IP + port)
- de l'adresse de transport de l'hôte distant (IP + port)

Une fois établie, une socket est donc identifiée par quatre données : deux adresses IP et deux ports.

Sockets

Plusieurs types de sockets selon l'usage :

- Pour le protocole TCP : socket stream.
- Pour le protocole UDP : socket datagram.
- Pour l'accès à la couche réseau : socket raw (pas de protocole de couche transport).

En TP, on travaillera avec le premier type.

Sockets : implémentation

- Différentes implémentations selon les systèmes d'exploitation : BSD sockets pour Linux.
- Des normes sont définies (POSIX : ensemble de normes pour les systèmes d'exploitation de type UNIX).
- API (Application Programming Interface) définies.
- java.net.Socket en Java, module Socket en Python.

TP Sockets

Lors des deux séances de TP à venir :

- Prise en main de l'API asyncio/stream (interface haut niveau avec les connexions réseaux).
- Programmation d'applications client ou serveur pour différents protocoles.
- Gestion de la concurrence (asyncio) pour les connexions en parallèle.

Démo asyncio

Programmation socket : client

- asyncio.run(client(server))
- reader, writer = await asyncio.open_connection(server, port)
- writer.write(message.encode())

  data = await reader.read()

Programmation socket : serveur

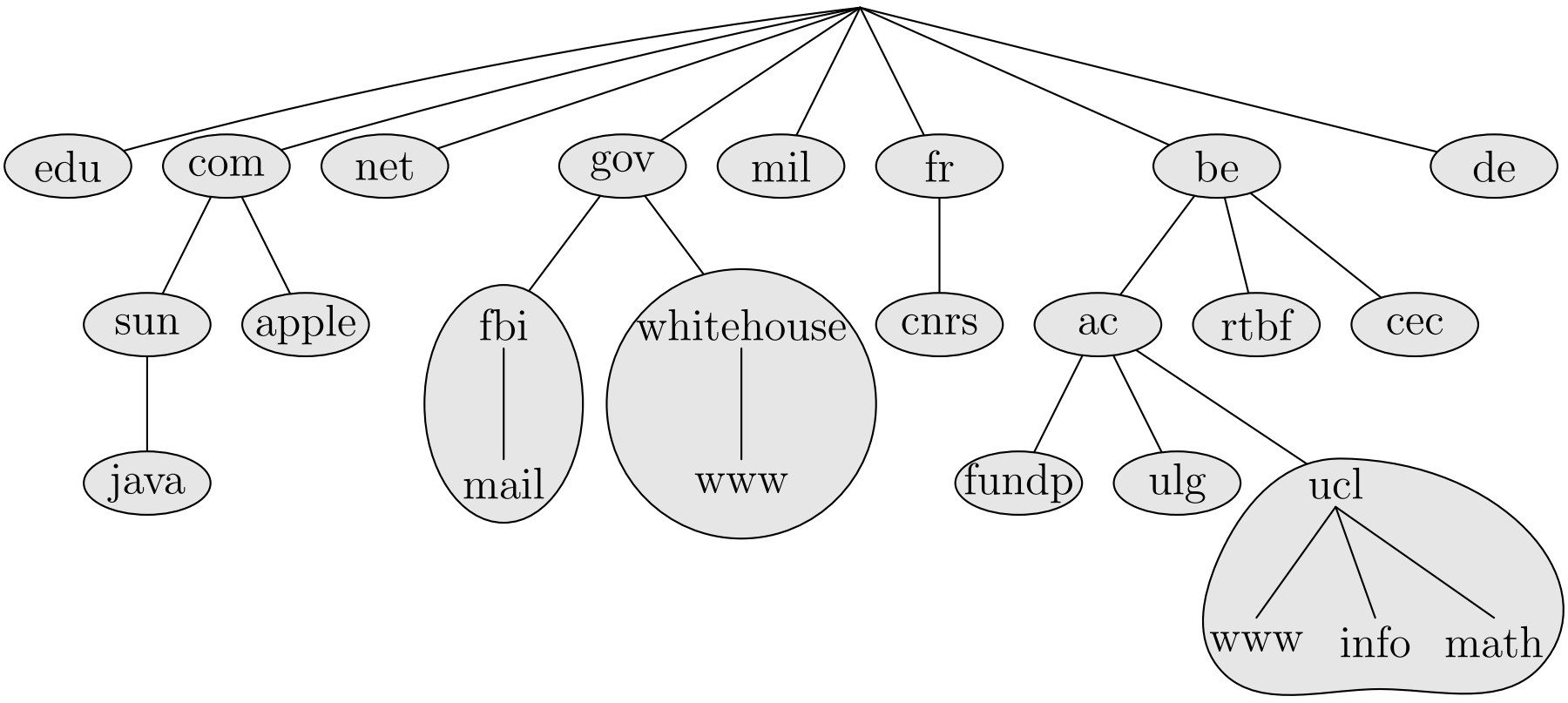
- asyncio.run(server())
- server = await asyncio.start_server(callback, servername, port)

  await server.serve_forever()
- async def callback(reader, writer) :

  ...
- data = await reader.readline()

  writer.write(message.encode())

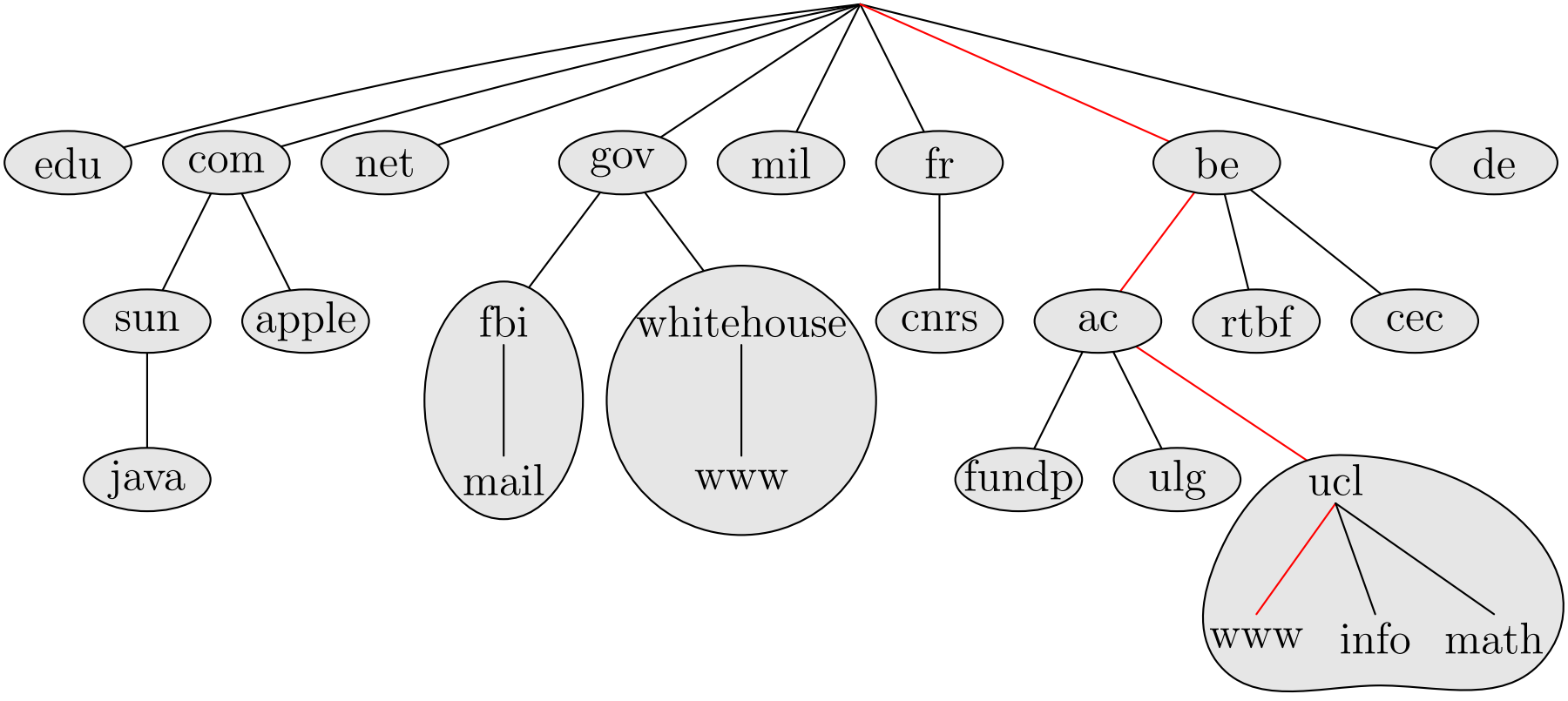
Domain Name System



Adresses humaines

Le but du DNS est d'établir le lien entre adresses IP et noms d'hôtes ou de domaines.

- C'est une base de données répartie entre des serveurs dédiés à cette tâche.
- Les noms sont organisés en un arbre dont chaque sous-arbre est un domaine.
- Chaque noeud a un nom, par exemple *ucl.ac.be*
- Chaque feuille correspond à un hôte (une machine), par exemple *www.ucl.ac.be*


edu
com
net
gov
mil
fr
be
de
sun
apple
fbi
whitehouse
cnrs
ac
rtbf
cec
java
mail
www
fundp
ulg
ucl
www
info
math

www.ucl.ac.be

IP : Rappels du cours 4

Une adresse IP est de la forme x.y.z.t où x, y, z et t sont entre 0 et 255.

- Si on oublie les détails, une adresse de réseau contient un préfixe qui représente un domaine et un suffixe laissé à 0, ex : 194.167.30.0
- Une adresse d'hôte dans ce réseau spécifie le suffixe, ex : 194.167.30.130
- L'ensemble des adresses a une structure d'arbre (premier préfixe 0.0.0.0 comme racine) avec à ses noeuds les sous-réseaux et aux feuilles les hôtes.

DNS fait le lien entre les deux arborescences.

Principe

- L'ensemble du réseau est découpé en zones.
- Chaque noeud appartient à exactement une zone.
- Pour chaque zone, il existe un ou plusieurs serveurs DNS officiels ou faisant autorité.
- Ce serveur officiel gère tous les noeuds de la zone et connaît les serveurs faisant autorité sur les zones correspondant à des sous-domaines.
- Pour connaître une adresse IP, il faut suivre le chemin depuis un serveur racine.

Requête DNS

- On appelle résolution DNS le fait d'obtenir la correspondance entre un nom de domaine et une adresse IP.
- La résolution est initiée par une requête DNS adressée par un client à un solveur DNS : chaque FAI possède un solveur DNS et fournit cette adresse lors de la connexion du client.
- Le solveur DNS obtient l'adresse et la fournit au client.
- Envoi de la requête en UDP sur le port 53 du solveur.

Résolutions DNS

Il existe 2 modes de résolution DNS :

- Requête récursive : Le serveur qui reçoit la requête fait la résolution et répond. Un serveur qui accepte les requêtes récursives est appelé *serveur récursif*. Ce sont les solveurs DNS des FAI en particulier.
- Requête itérative : Le serveur qui reçoit la requête répond soit avec l'information demandée s'il peut, soit avec un lien vers un serveur plus proche auquel il délègue la responsabilité. Ce sont les serveurs faisant autorité sur une zone en particulier.

Chaque serveur DNS connaît le nom et l'adresse IP des serveurs auxquels il délègue.

Résolution DNS

- Le serveur qui a reçu la requête DNS répond directement s'il est responsable de l'adresse demandée.
- Sinon :
  - si c'est un serveur récursif, il interroge le serveur connu le plus proche.
  - si c'est un serveur itératif, il envoie en réponse un pointeur vers le serveur connu le plus proche.

Le serveur connu le plus proche étant :

- le serveur faisant autorité sur ce domaine si l'adresse appartient à un sous domaine.
- sinon, un serveur racine.

Serveurs DNS

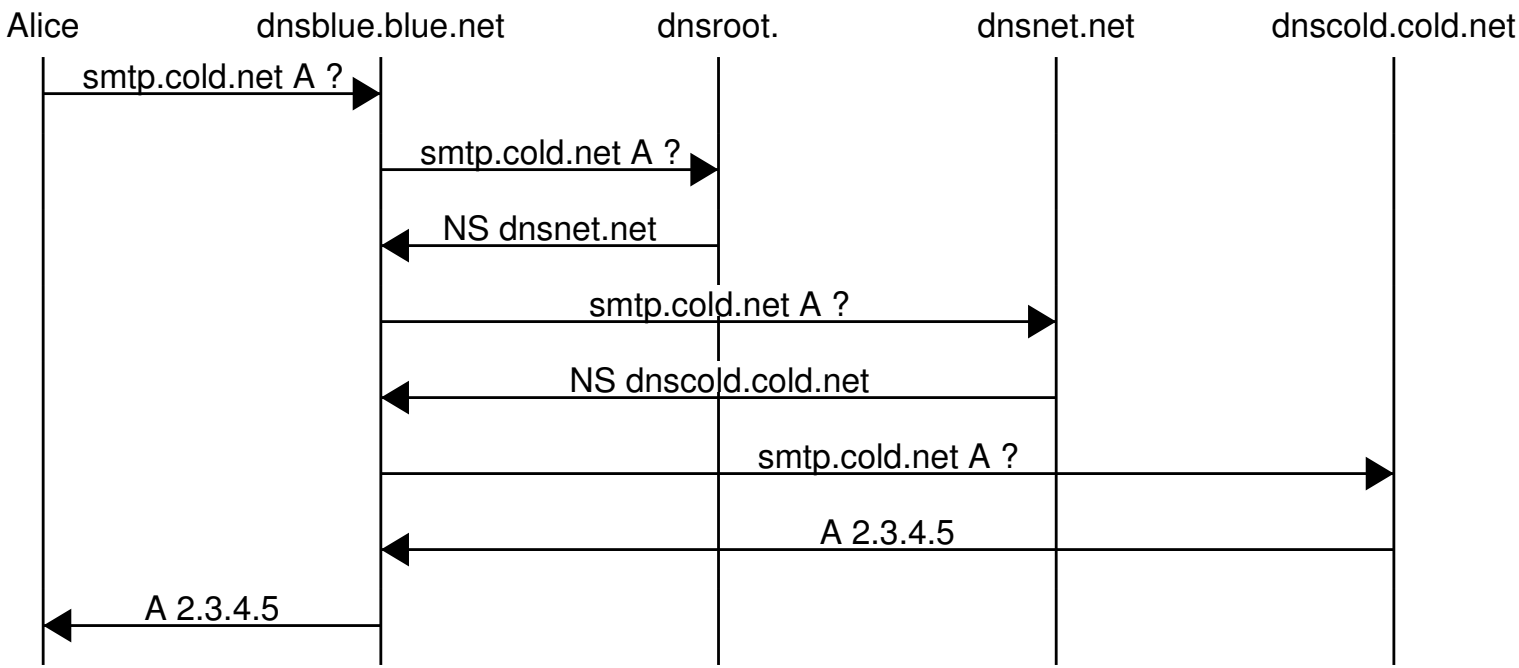
Différents types de serveurs DNS :

- Les serveurs racines qui connaissent l'adresse d'un serveur officiel pour chaque domaine de premier niveau (TLD : com, net, org...)
- Chaque serveur DNS doit connaître l'adresse d'un serveur racine pour débuter la résolution.
- Les serveurs de zones (TLD ou non) qui connaissent les adresses des hôtes de leur zone et les adresses d'un serveur officiel dans chaque sous-domaine.

Résolution DNS

- Le client envoie une requête DNS récursive au serveur DNS **F** de son FAI.
- Ce serveur **F**, s'il ne connaît pas l'adresse demandée, envoie une requête itérative à un serveur racine **A** qui lui répond avec l'adresse d'un serveur DNS **B** auquel il délègue la responsabilité.
- Le serveur **F** poursuit en envoyant une requête itérative à **B**, etc...

Exemple


Alice
dnsblue.blue.net
dnsroot.
dnsnet.net
dnscold.cold.net
smtp.cold.net A ?
smtp.cold.net A ?
NS dnsnet.net
smtp.cold.net A ?
NS dnscold.cold.net
smtp.cold.net A ?
A 2.3.4.5
A 2.3.4.5

Robustesse et temps d'exécution

Pour optimiser le fonctionnement du service :

- L'information est dédoublée : chaque nom de domaine est connu par au moins deux serveurs.
  primary name et secondary name servers.
- Chaque réponse reçue par un serveur est conservée en cache.
  Time To Live (TTL) fixé par le serveur faisant autorité.

Resource Records

Un enregistrement DNS (dans le fichier de configuration d'un serveur DNS) contient :

Nom	TTL	Classe	Type	Valeur

- Nom : Nom de domaine concerné
- TTL : Time To Live de l'enregistrement (vide si valeur par défaut)
- Classe : En général IN pour internet
- Type : Type d'enregistrement
- Valeur : Selon le type

Types d'enregistrement

- A : Adresse IPv4
- AAAA : Adresse IPv6
- MX : Mail eXchanger (avec une priorité)
- TXT : Commentaires
- NS : Name Server (nom du serveur à contacter)
- SOA : Start Of Authority (informations sur la zone DNS)
- CNAME : Canonical NAME (pour un alias)

Exemple dans labobs (dnscom)

Pour la zone com.

```
$TTL    60000
@               IN    SOA   dnscom.com.  root.dnscom.com. (
                      1  ; serial
                      28  ; refresh
                      14  ; retry
                      3600000  ; expire
                      60000  ; negative cache ttl
                      )
@               IN    NS    dnscom.com.
dnscom          IN    A     42.13.37.42

jmail           IN    NS    dnsjmail.jmail.com.
dnsjmail.jmail  IN    A     173.194.66.108

perdu           IN    NS    dnsperdu.perdu.com.
dnsperdu.perdu  IN    A     8.8.8.8
```

Exemple dans labobs (perdu)

Pour la zone perdu.com.

```
$TTL    60000
@           IN    SOA    dnsperdu.perdu.com.  root.dnsperdu
                  1  ; serial
                  28  ; refresh
                  14  ; retry
                  3600000  ; expire
                  60000  ; negative cache ttl
                  )
@           IN    NS    dnsperdu.perdu.com.
dnsperdu    IN    A     8.8.8.8
www         IN    A     8.8.8.8
```

```
% dig -t ANY univ-orleans.fr
; <<>> DiG 9.11.4-4-Debian <<>> -t ANY univ-orleans.fr
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52661
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 10, AUTHORITY: 0, ADDITIONAL: 13

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;univ-orleans.fr.               IN      ANY

;; ANSWER SECTION:
univ-orleans.fr.        86400   IN      TXT     "v=spf1 mx ip4:194.167.30.0/24 ~all"
univ-orleans.fr.        86400   IN      SOA     dns.univ-orleans.fr. gestion-dns.univ-orleans.fr. 20180913
univ-orleans.fr.        86400   IN      NS      dns.univ-orleans.fr.
univ-orleans.fr.        86400   IN      NS      dns2.univ-orleans.fr.
univ-orleans.fr.        86400   IN      NS      dns3.univ-orleans.fr.
univ-orleans.fr.        86400   IN      MX      50 mxc.relay.renater.fr.
univ-orleans.fr.        86400   IN      MX      50 mxd.relay.renater.fr.
univ-orleans.fr.        86400   IN      MX      50 mxa.relay.renater.fr.
univ-orleans.fr.        86400   IN      MX      50 mxb.relay.renater.fr.
univ-orleans.fr.        86400   IN      A       194.167.30.239

;; ADDITIONAL SECTION:
dns.univ-orleans.fr.    86400   IN      A       194.167.30.130
dns.univ-orleans.fr.    86400   IN      AAAA    2001:660:6402:1200::130
dns2.univ-orleans.fr.   86400   IN      A       194.167.31.3
dns3.univ-orleans.fr.   86400   IN      A       194.167.31.100
mxa.relay.renater.fr.   2248    IN      A       194.214.201.8
mxa.relay.renater.fr.   1207    IN      AAAA    2001:660:3027:1::8
mxb.relay.renater.fr.   2248    IN      A       194.214.201.9
mxb.relay.renater.fr.   1207    IN      AAAA    2001:660:3027:1::9
mxc.relay.renater.fr.   2090    IN      A       194.214.200.8
mxc.relay.renater.fr.   1207    IN      AAAA    2001:660:3027:2::8
mxd.relay.renater.fr.   2248    IN      A       194.214.200.9
mxd.relay.renater.fr.   1207    IN      AAAA    2001:660:3027:2::9
```

```
% dig -t ANY www.univ-orleans.fr
; <<>> DiG 9.11.4-4-Debian <<>> -t ANY www.univ-orleans.fr
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49828
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 6

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.univ-orleans.fr.           IN      ANY

;; ANSWER SECTION:
www.univ-orleans.fr.    86400   IN      A       194.167.30.239
www.univ-orleans.fr.    86400   IN      MX      0 smtp.univ-orleans.fr.

;; AUTHORITY SECTION:
univ-orleans.fr.        86400   IN      NS      dns2.univ-orleans.fr.
univ-orleans.fr.        86400   IN      NS      dns.univ-orleans.fr.
univ-orleans.fr.        86400   IN      NS      dns3.univ-orleans.fr.

;; ADDITIONAL SECTION:
smtp.univ-orleans.fr.   86400   IN      A       194.167.30.100
dns.univ-orleans.fr.    86400   IN      A       194.167.30.130
dns.univ-orleans.fr.    86400   IN      AAAA    2001:660:6402:1200::130
dns2.univ-orleans.fr.   86400   IN      A       194.167.31.3
dns3.univ-orleans.fr.   86400   IN      A       194.167.31.100

;; Query time: 3 msec
;; SERVER: 192.168.80.232#53(192.168.80.232)
;; WHEN: mar. sept. 04 14:04:53 CEST 2018
;; MSG SIZE  rcvd: 233
```

Reverse DNS

DNS offre la possibilité de trouver le nom de domaine associé à une adresse IP :

- Création d'un domaine approprié : in-addr.arpa
- Transformation de l'adresse IP en nom de domaine dans in-addr.arpa :

  168.152.13.27 devient 27.13.152.168.in-addr.arpa

- Type supplémentaire PTR

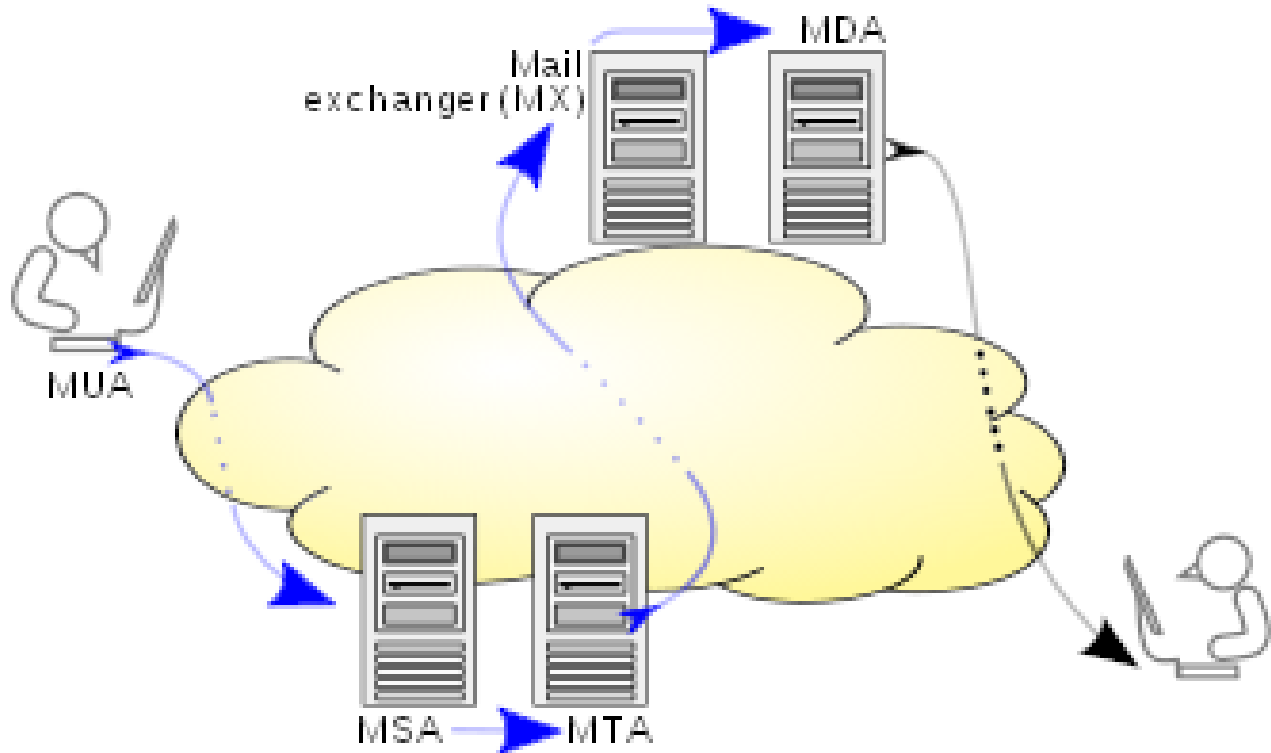
Reverse DNS : objectif

Le but principal est de permettre une vérification de certaines informations DNS :

- confirmer une réponse DNS avec une requête reverse DNS (manque de sécurité du protocole).
- lutter contre le SPAM en comparant :
  - le nom de domaine associé à l'adresse IP de l'émetteur ;
  - le nom de domaine de l'adresse mail annoncée comme expéditeur.

```
% dig -x 194.167.30.239
; <<>> DiG 9.11.4-4-Debian <<>> -x 194.167.30.239
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 44949
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 7

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;239.30.167.194.in-addr.arpa.   IN      PTR

;; ANSWER SECTION:
239.30.167.194.in-addr.arpa. 10800 IN   PTR     sulkut.univ-orleans.fr.

;; AUTHORITY SECTION:
30.167.194.in-addr.arpa. 10800  IN      NS      arcturus.ciril.fr.
30.167.194.in-addr.arpa. 10800  IN      NS      dns2.univ-orleans.fr.
30.167.194.in-addr.arpa. 10800  IN      NS      dns.univ-orleans.fr.
30.167.194.in-addr.arpa. 10800  IN      NS      dns3.univ-orleans.fr.

;; ADDITIONAL SECTION:
dns.univ-orleans.fr.    86400   IN      A       194.167.30.130
dns.univ-orleans.fr.    86400   IN      AAAA    2001:660:6402:1200::130
dns2.univ-orleans.fr.   86400   IN      A       194.167.31.3
dns3.univ-orleans.fr.   86400   IN      A       194.167.31.100
arcturus.ciril.fr.      12116   IN      A       193.50.27.66
arcturus.ciril.fr.      12116   IN      AAAA    2001:660:4503:201::66

;; Query time: 1 msec
;; SERVER: 192.168.80.232#53(192.168.80.232)
;; WHEN: mar. sept. 04 14:44:47 CEST 2018
;; MSG SIZE  rcvd: 297
```

Simple Mail Transfer Protocol



Les principes

- Envoi de messages désynchronisé.
- Diversité de formats, ASCII à l'origine, généralisation à des binaires (son, image, formats divers) avec MIME (Multipurpose Internet Mail Extensions).
- Utilisation du DNS pour trouver le serveur de destination.

Le chemin

- Mail User Agent : Client mail (alpine, Thunderbird,. . .), webmail
- Mail Transfer Agent : Transfert vers un autre serveur
- Mail Delivery Agent : Dépôt dans une boîte mail

Deux types de protocole :

- Pour envoyer un message SMTP, TCP port 25, RFC 821
- Pour récupérer les messages de la boîte mail POP, IMAP

MSA & MTA

On a tendance a dissocier les rôles de Mail Submission Agent et MTA. Les deux reçoivent des données lors d'une session SMTP, mais :

- le MSA n'accepte que des mails venant du MUA (SMTP en général sur le port 587), donc des mails sortants ;
- alors que le MTA n'accepte que des mails venant d'un autre MTA, donc des mails entrants.

Dans ce cas, MSA et MTA collaborent côté émetteur, symétriquement à MTA et MDA côté receveur.

Le contenu

Plusieurs entêtes puis le corps après une ligne vide. Entêtes fortement recommandées :

- To : mdelacourt@dim.uchile.cl
- From :martin.delacourt@univ-orleans.fr
- Date : Tue, 4 Sep 2018 15 :21 :45 +0200

D'autres moins :

- Subject : Hola
- cc : martin.delacourt@univ-lorraine.fr
- MIME-Version : 1.0
- Content-Type : ext/plain ; charset=utf-8 ;

```
Return-Path: <martin.delacourt@univ-orleans.fr>
X-Original-To: mdelacourt@dim.uchile.cl
Delivered-To: mdelacourt@dim.uchile.cl
Received: from scfilter.dim.uchile.cl (unknown [10.0.0.179])
        (using TLSv1.2 with cipher ECDHE-RSA-AES256-GCM-SHA384 (256/256 bits))
        (No client certificate requested)
        by gauss.dim.uchile.cl (Postfix) with ESMTPS id 93FDAA5DDB9
        for <mdelacourt@dim.uchile.cl>; Tue,  4 Sep 2018 12:31:57 -0300 (-03)
Received: by scfilter.dim.uchile.cl (Postfix, from userid 65534)
        id 424W6543FlzyVj; Tue,  4 Sep 2018 12:31:57 -0300 (-03)
Received: from localhost (localhost.localdomain [127.0.0.1])
        by scfilter.dim.uchile.cl (Postfix) with ESMTP id 424W650MRQzyVL
        for <mdelacourt@dim.uchile.cl>; Tue,  4 Sep 2018 12:31:57 -0300 (-03)
X-Virus-Scanned: Scrollout F1 at dim.uchile.cl
X-Spam-Flag: NO
X-Spam-Score: 1.504
X-Spam-Level: *
X-Spam-Status: No, score=1.504 tagged_above=-1000 required=5
        tests=[BAYES_00=-1.9, DATE_IN_PAST_96_XX=3.405,
        RCVD_IN_DNSWL_NONE=-0.0001, SPF_PASS=-0.001]
        autolearn=no autolearn_force=no, No
Received: from scfilter.dim.uchile.cl ([127.0.0.1])
        by localhost (scfilter.dim.uchile.cl [127.0.0.1]) (amavisd-new, port 10024)
        with LMTP id ks6lD8C9eGCb for <mdelacourt@dim.uchile.cl>;
        Tue,  4 Sep 2018 12:31:42 -0300 (-03)
Received: from dio.univ-orleans.fr (dio.univ-orleans.fr [194.167.30.47])
        by scfilter.dim.uchile.cl (Postfix) with ESMTP id 424W5n5NdpzyV5
        for <mdelacourt@dim.uchile.cl>; Tue,  4 Sep 2018 12:31:41 -0300 (-03)
Received: from localhost (localhost [127.0.0.1])
        by dio.univ-orleans.fr (Postfix) with ESMTP id DFE0E7AB0D
        for <mdelacourt@dim.uchile.cl>; Tue,  4 Sep 2018 17:31:25 +0200 (CEST)
X-Virus-Scanned: Debian amavisd-new at dio.univ-orleans.fr
Received: from dio.univ-orleans.fr ([127.0.0.1])
        by localhost (dio.univ-orleans.fr [127.0.0.1]) (amavisd-new, port 10024)
        with ESMTP id DFogT9XUXNX6 for <mdelacourt@dim.uchile.cl>;
        Tue,  4 Sep 2018 17:31:24 +0200 (CEST)
```

```
Received: from Chao (unknown [192.168.80.74])
        by dio.univ-orleans.fr (Postfix) with SMTP id CECC47AB0B
        for <mdelacourt@dim.uchile.cl>; Tue,  4 Sep 2018 17:29:46 +0200 (CEST)
From: Martin Delacourt <martin.delacourt@univ-orleans.fr>
To: mdelacourt@dim.uchile.cl
Date: Tue, 4 Sep 2010 15:21:45 +0200
Subject: Hola!
Message-Id: <20180904153125.DFE0E7AB0D@dio.univ-orleans.fr>
X-UCHILE-MailScanner-Information: Please contact the ISP for more information
X-UCHILE-MailScanner-ID: 93FDAA5DDB9.A9A67
X-UCHILE-MailScanner: Found to be clean
X-UCHILE-MailScanner-From: martin.delacourt@univ-orleans.fr

Que tal?
```

Le protocole

- Protocole textuel, facile à comprendre (alice@jmail.com écrit à bob@cold.net).
- Le MUA d'Alice transmet le message au serveur de courrier sortant de jmail.com, au moyen d'une session SMTP.
- Le serveur sortant résout l'enregistrement DNS de type MX pour cold.net.
- Il transmet le message au moyen d'une nouvelle session SMTP.
- Le serveur entrant de cold.net stocke le message dans la boîte mail de Bob.

Pour chaque session SMTP, le client demande une connexion TCP avec le serveur, celui-ci lance alors l'échange.

nc dio.univ-orleans.fr 25

220 dio.univ-orleans.fr ESMTP Postfix (Debian/GNU)

HELO Chao

250 dio.univ-orleans.fr

MAIL FROM: martin.delacourt@univ-orleans.fr

250 2.1.0 Ok

RCPT TO: mdelacourt@dim.uchile.cl

250 2.1.5 Ok

DATA

354 End data with <CR><LF>.<CR><LF>

From: Martin Delacourt <martin.delacourt@univ-orleans.fr>
To: Martin Delacourt <mdelacourt@dim.uchile.cl>
Date: Tue, 14 Dec 1979 15:21:45 +0200
Subject: Hola!

Que tal?

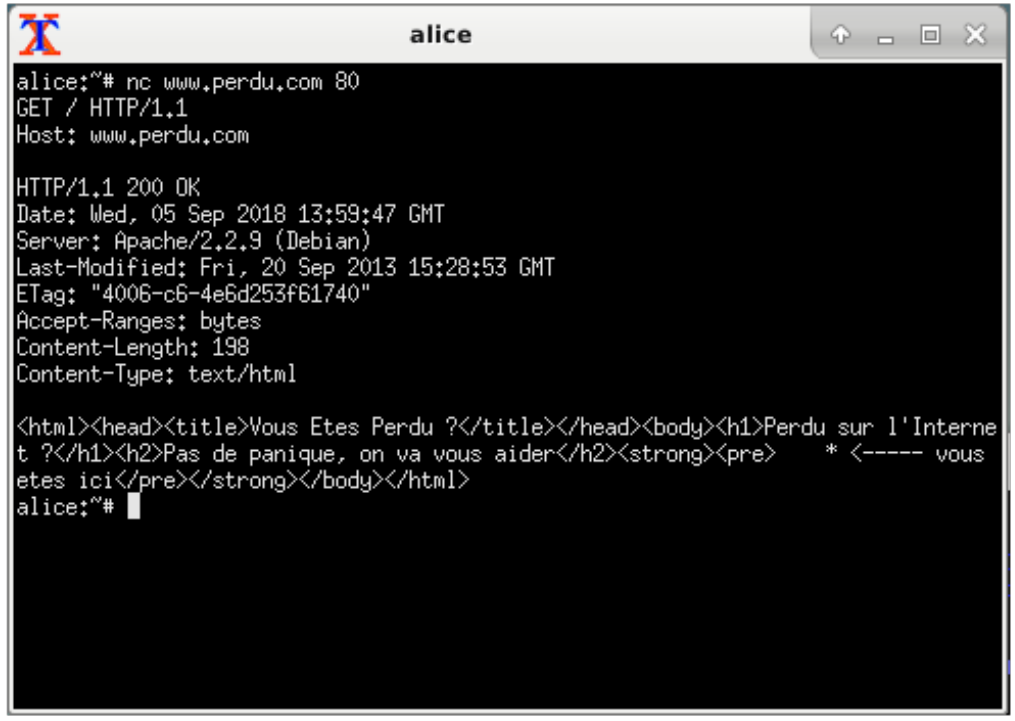
.

250 2.0.0 Ok: queued as CECC47AB0B

QUIT

221 2.0.0 Bye

HyperText Transfer Protocol



Principe

Récupérer des documents hypertexte stockés sur un serveur web :

- Client : navigateur.
- Résolution du nom de domaine passé dans l'url : champ DNS de type A.
- Un format de fichier : HTML.
- Un protocole d'échanges : HTTP

Uniform Resource Locator

(RFC 3986) Identifiant unique d'une ressource contenant
le moyen d'y accéder, l'hôte et le chemin sur l'hôte :

`http://www.perdu.com`

`https://celene.univ-orleans.fr/course/view.php?id=1980`

`http://cnp3book.info.ucl.ac.be/1st/html/application/http.html`

`file:///etc/apt/sources.list`

`ftp://ftp.lip6.fr/pub/rfc/rfc/rfc1149.txt.gz`

HyperText Markup Language

```
<html>
  <head>
    <title>Vous Etes Perdu  ?</title>
  </head>
  <body>
    <h1>Perdu sur l'Internet  ?</h1>
    <h2>Pas de panique, on va vous aider</h2>
    <strong>
      <pre>    * <----- vous &ecirc ;tes ici</pre>
    </strong>
  </body>
</html>
```

HyperText Transfer Protocol

- protocole textuel, TCP, port 80.
- Actuellement HTTP 1.1, dans les RFC 7230 à 7237.
  Et HTTP 2 (RFC 7540).
-
  - Commande GET suivi du chemin du document et du protocole
  - puis Host: sur une nouvelle ligne
  - puis une ligne vide.

nc -C www.perdu.com 80

GET / HTTP/1.1
Host: www.perdu.com

HTTP/1.1 200 OK
Date: Tue, 04 Sep 2018 16:01:18 GMT
Server: Apache
Last-Modified: Thu, 02 Jun 2016 06:01:08 GMT
ETag: "cc-5344555136fe9"
Accept-Ranges: bytes
Content-Length: 204
Vary: Accept-Encoding
Content-Type: text/html
<html><head><title>Vous Etes Perdu
?</title></head><body><h1>Perdu sur l'Internet
?</h1><h2>Pas de panique, on va vous
aider</h2><pre> * <—— vous êtes
ici</pre></body></html>

Droits

L'image utilisée en page 39 provient de wikimedia commons.

`http://commons.wikimedia.org/wiki/File:SMTP-transfer-model.svg`

Mes remerciements à Nicolas Ollinger (Université d'orléans) pour ses contributions.