

INTERACTION DISTRIBUÉE

Ph. Truillet

https://www.github.com/truillet

Septembre 2023



UN MONDE EN RÉSEAU

avant-hier

internet connecte tous les ordinateurs (ou presque)

hier

 les terminaux interactifs sont omniprésents (notion d'informatique embarquée): smartphones, tablettes, ...

aujourd'hui et demain

 chaque objet physique <u>est</u> connecté (notion informatique diffuse ou ubiquitaire – pervasive computing, « internet of things » IoT, smart cities, …)





UN PEU D'HISTOIRE

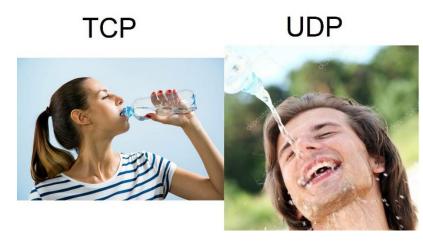
En 1982, « The internet Coke Machine », une machine à boissons de Carnegie Mellon University est connectée à des terminaux via une liaison série (Michael Kazar) : elle fait un rapport de son stock et donne sa température sur le réseau.

En **1992**, la « *Trojan Room Coffee Pot* » à Cambridge University (fermée le 22/08/2001)





RÉSEAUX



Quand on parle de réseaux, on l'associe souvent à un protocole réseau généralisé depuis les années 90 -> la pile de protocoles **TCP/IP**

Néanmoins, on a :

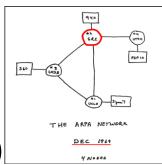
- des couches physiques hétérogènes
- des protocoles et des architectures <u>hétérogènes</u>

Et bien évidemment, de nombreux protocoles non-IP (LoRa, SigFox, Zigbee, ...)!



The augmentation of Doug Engelbart - https://www.dougengelbart.org

Un des pionniers : Douglas Engelbart (1925-2013)



- A l'intuition d'internet¹ dès les années 50 (son laboratoire (SRI)
 participe à la première liaison en 1969 avec l'UCLA)
- Démontre la première vidéoconférence (1968) « The Mother of All demos »
- Invente la souris (1968)

http://www.dougengelbart.org/pubs/augment-3906.html

²The Mother of All demos,

http://www.dougengelbart.org/firsts/dougs-1968-demo.html





¹Augmented Human intellect:

D'autres pionniers:

Louis Pouzin (1931-?), « inventeur » de la commutation de paquets (IRIA, Projet Cyclades 1971-1978)

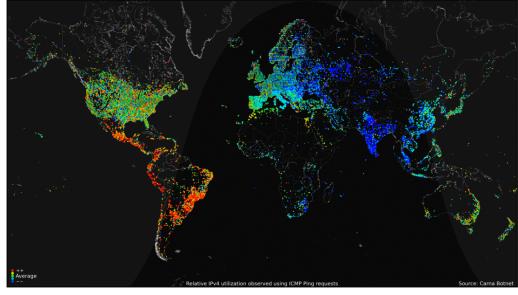


• Vint Cerf (1943-?), « co-inventeur » du protocole TCP/IP, « Chief Internet Evangelist » chez Google depuis 2005



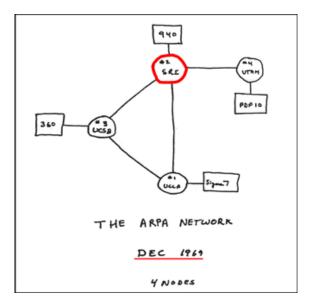
•





http://motherboard.vice.com/blog/this-is-most-detailed-picture-internet-ever

Internet est ... un réseau de réseaux (hétérogènes)



2900767	2100	CONDED OP. PROGRAM	SK
		CONDED OP PROGRAM EDIZ BEN BARKER BBV	
	22:30		cSo
		talked to SRI Host to Host	
		Cefto inp grogram	CSL
		a host lend message	
		to imp.	1

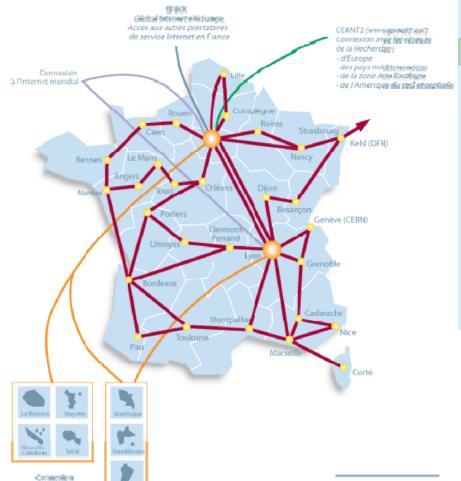
UNIVERSITÉ TOULOUSE III PAUL SABATIER



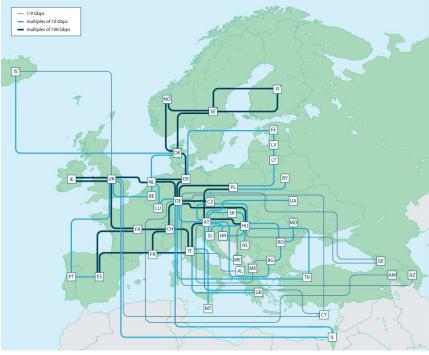
Infrastructure du réseau

syns, les, £35%-1566.

CONNECTEUR DE SAVOIRS



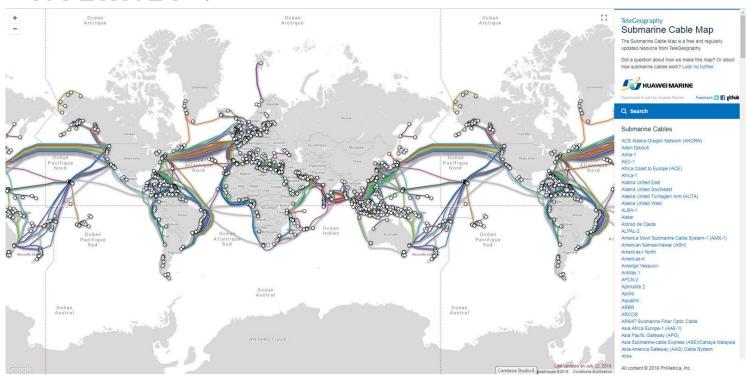








Infrastructure #ENATERAF



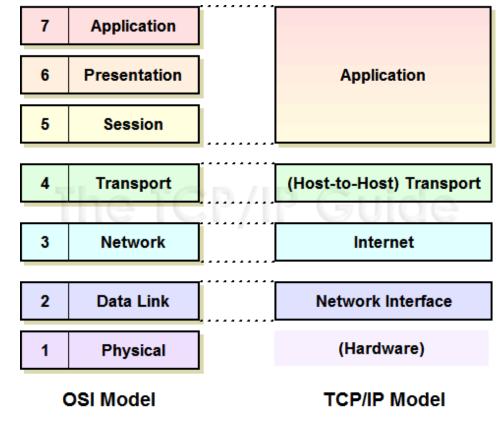
https://www.submarinecablemap.com

Câble le plus long : 39 000 km (SEA ME WE 3)

Au total, 800 000 km (20 fois le tour de la terre) et 99% des communications mondiales



internet ... est un ensemble de protocoles (1969/1972)



http://www.tcpipguide.com/free



Wint is your address:



mean, your physical 00:A0:C9:4F:73:2E

IP "Internet Protocol" (couche réseau)

échange les données entre ordinateurs hôtes

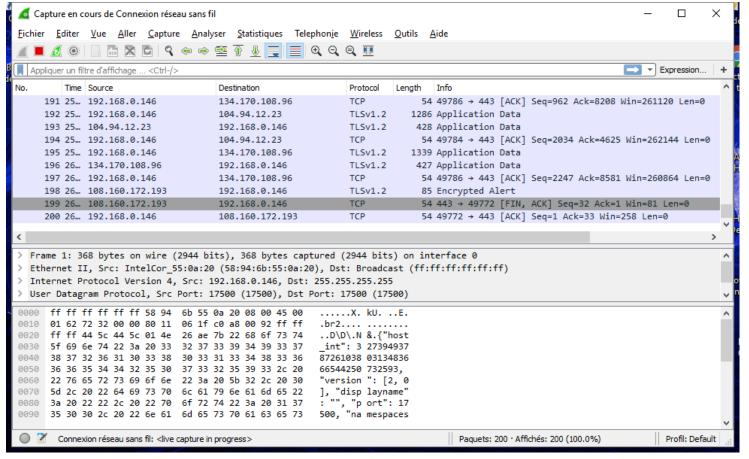
TCP "Transport Control Protocol" (couche transport)

échange les données entre les applications



INTERNET ...?

Ce fonctionnement « en couches » est intéressant mais pose certains de nombreux problèmes de sécurité.



http://www.wireshark.org



INTERNET ...?

La plupart des protocoles utilisés (et pas que dans ce cours !) sont forgés sur **ces couches** et vont permettre de faire une abstraction plus ou moins importante du réseau !

Thread

iamkirkbater and jkjustjoshing



iamkirkbater Mar Aug 23rd, 2017 at 9:37 AM in #www

Do you want to hear a joke about TCP/IP?



7 replies



jkjustjoshing 5 months ago

Yes, I'd like to hear a joke about TCP/IP



iamkirkbater 📓 5 months ago

Are you ready to hear the joke about TCP/IP?



jkjustjoshing 5 months ago

I am ready to hear the joke about TCP/IP



iamkirkbater 📓 5 months ago

Here is a joke about TCP/IP.



iamkirkbater 🚨 5 months ago

Did you receive the joke about TCP/IP?



jkjustjoshing 5 months ago

I have received the joke about TCP/IP.



iamkirkbater 3 months ago

Excellent. You have received the joke about TCP/IP. Goodbye.



SYSTÈMES RÉPARTIS

Une définition très large: un système réparti est un système informatique dans lequel les <u>ressources</u> ne sont pas centralisées

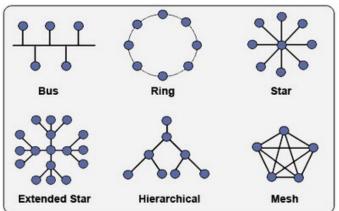
Les ressources ont un sens très large :

- stockage (disques, bases de données)
- puissance de calcul
- mais aussi les utilisateurs



SYSTÈMES RÉPARTIS

- But : permettre à des utilisateurs de manipuler (calculer) leurs données (stocker) sans contrainte sur les localisations respectives des éléments du système
- La plupart du temps, cela correspond à la généralisation et l'amélioration du schéma client/serveur :
 - serveurs multiples (équilibrage de charge, redondance)
 - systèmes multi-couches (tiers)
 - peer to peer
- Pour ce que l'on verra → réparti ≈ <u>distribué</u>





POURQUOI DES SYSTÈMES RÉPARTIS ?

- pour des aspects économiques
 - réalisation de systèmes à haute disponibilité
 - partage de ressources (programmes, données, services)
 - réalisation de systèmes à grande capacité d'évolution
- pour une adaptation de la structure d'un système à celle des applications
- pour un besoin d'intégration
- pour un besoin de communication et de partage d'information



LES DOMAINES D'APPLICATION

ingénierie

- coopération d'équipes pour la conception d'un produit
- production coopérative de documents
- partage cohérent d'information

gestion intégrée des informations d'une entreprise

intégration de l'existant

contrôle et organisation d'activités en temps réel centres de documentation, bibliothèques

recherche, navigation, visualisation multimédia

systèmes d'aide à la formation (collecticiels, ...)



LES BESOINS DES APPLICATIONS

ouverture

· interopérabilité, portabilité, fédération ; réutilisation de l'existant

coopération, coordination, partage

- vision commune cohérente d'informations partagées (globalement, par groupes)
- interaction en temps réel, support multimédia

transparence

- accès (mobilité des usagers avec préservation de l'environnement)
- localisation (de l'information, des services, ...)



LES BESOINS DES APPLICATIONS

qualité de service

disponibilité, délais, coûts, qualité de perception, ... avec niveau garanti

sécurité

authentification, intégrité, confidentialité, ...

évolutivité, administrabilité

reconfiguration, gestion dynamique de services



QUELQUES DIFFICULTÉS LIÉES À LA RÉPARTITION

- l'accès aux données distantes
- le maintien du service (notamment la gestion des pannes)

l'encodage/décodage des données (marshalling, unmarshalling)

- les accès concurrents au(x) service(s)
- la gestion des droits à distance



NOTIONS FONDAMENTALES: CLIENT/SERVEUR ET PROTOCOLES



QUELQUES NOTIONS ARCHITECTURE CLIENT/SERVEUR

Serveur: celui qui offre un service (doit l'offrir de manière permanente) \rightarrow « daemon »

- Accepte les requêtes, les traite en renvoie une réponse
- Ex : httpd, ftpd, telnetd, ...

Client : celui qui utilise le service

Envoie une requête et reçoit la réponse





QUELQUES NOTIONS ARCHITECTURE CLIENT/SERVEUR

Architecture C/S \rightarrow description du <u>comportement coopératif</u> entre

le serveur et les clients

- → fonctionnement général des services internet
- s'appuie sur des <u>protocoles</u> entre les processus communicants



QUELQUES NOTIONS PROTOCOLE

On nomme **protocole** les **conventions** qui facilitent une communication sans faire directement partie du sujet de la communication elle-même

Exemples de protocole : FTP, HTTP, ...

Sont compilées dans les RFC (**R**equest **f**or **C**omments). Peu de RFC sont des normes mais toutes les normes internet sont enregistrées en tant que RFC

- FTP \rightarrow RFC 114 (Avril 1971), HTTP \rightarrow RFC 1945 (mai 1996), RFC 2616 (juin 1999), ...
- Les RFC du 1er avril:
 - IPoAC (RFC 1149, RFC 6214 pour IPv6) ☺
 - HTJP HyperText Jeopardy Protocol (RFC 8565)
 - TCP Option to Denote Packet Mood (RFC 5841)
 - ...



QUELQUES NOTIONS PROTOCOLE

HTTP 1.1

Ex: HTTP/1.1 400 Bad Request

Date: Tue, 13 Sep 2011 14:27:22 GMT

Server: Apache

Content-Length: 226

Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">



QUELQUES NOTIONS SERVICES

Une machine offre <u>usuellement</u> plusieurs services accessibles par un numéro de <u>port</u> (de 1 à 65535)

On doit connaître ce numéro pour accéder au service → notion de ports
 « bien connus »

Ex: echo: port 7

daytime: port 13

ftp: port 21

http: port 80

/etc/services sous Linux



EXEMPLE: LE DNS

DNS: Domain Name Server (1984)

Permet de trouver l'adresse IP correspondant au nom de domaine

exemple:sri.univ-tlse3.fr \rightarrow 195.220.43.54

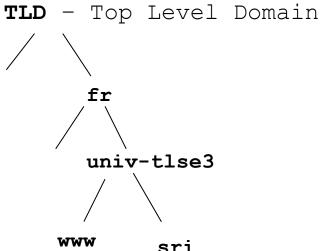
DNS: base de donnée répartie, système hiérarchique



EXEMPLE: DNS

un serveur DNS gère un domaine

le gestionnaire peut déléguer la gestion d'un sous-domaine à une autre gestionnaire

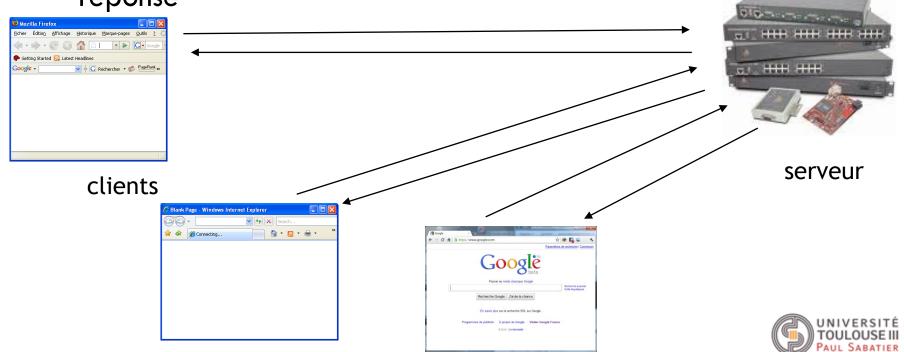




EXEMPLE: HTTP

HTTP: HyperText Transfer Protocol

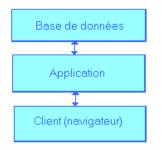
Protocole client-serveur très simple. Une requête \rightarrow une réponse



EXEMPLE: ARCHITECTURE MULTI-COUCHES

Classiquement, 3 couches (extension du modèle Client/Serveur)

- Présentation : (Interface) visible et interactive
- Application : (partie fonctionnelle) couche métier, logique applicative
- Stockage



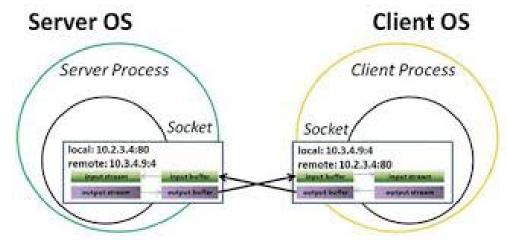


EXEMPLE: P2P

Principe du « pair-à-pair » : chacun est <u>à la fois</u> client et serveur

- p2p « pur »: connexions directes entre participants
- p2p « pratique » : des serveurs existent entre les clients permettant l'existence d'un service d'annuaire (qui est connecté, qui propose quoi et où ?, ...)





Plusieurs niveaux d'abstraction :

Bas-niveau : la socket (échanges de messages)

```
// connexion
laSocket = new Socket(machine, port);

// Mise en forme du flux de sortie
fluxSortieSocket = new
    PrintStream(laSocket.getOutputStream());

fluxSortieSocket.println("GET /cam picture HTTP/1.0\r\n");
```



Appel à des <u>procédures</u> distantes (RPC, SOAP, ...)

```
Resultat = calcul_addition_1(&parametre, clnt);

if (resultat == (reponse *) NULL) {
  clnt_perror (clnt, "call failed");
  clnt_destroy (clnt);
  exit(EXIT_FAILURE);
}
```



Appel à des <u>méthodes</u> distantes (RMI, CORBA, ...)



Déclenchement <u>d'événements</u> distants (OSGi, ...)

```
public class AmpouleListener implements ServiceListener {
  public void serviceChanged(ServiceEvent event) {
      // ServiceReference ref = e.getServiceReference();
      String[] objectClass = (String[])
      event.getServiceReference().getProperty("objectClass");
```





Publisher/Subscriber (MQTT, ROS, ...)

```
cli = paho.Client(client id="PiZero2")
cli.on message = on message
cli.on publish = on publish
cli.username pw set("try", password="try")
cli.connect("broker.shiftr.io", 1883, 60);
cli.subscribe("/data",0);
while cli.loop() == 0:
 time.sleep(60) # delay for 1 minute
```



CE QUE L'ON VA FAIRE

- des travaux ... pratiques (4 +2 séances)!
- programmation en java et python (la plupart du temps) mais vous pouvez choisir un autre langage si vous le souhaitez

Objectifs:

- acquérir des notions solides sur les abstractions réseau les plus utilisés.
- tester les avantages et inconvénients de chaque approche



(PETITS) OUTILS

PuTTY, KiTTY:

http://www.9bis.net/kitty/?zone=fr



? httpie

https://httpie.io



http://curl.haxx.se

Terminal:

https://sites.google.com/site/terminalbpp



TRAVAUX PRATIQUES (4+2 SÉANCES)

- Echange de messages
 - ZeroMQ / ingescape (4 x 2h)



- bus à événements, prototypage ivy >> utile pour le TP IHM
- MQTT 2 h (+ raspberry)



- Appel de méthodes et d'objets à distance
 - JSON / API REST 2 h



(Anciennes ressources disponibles)



EVALUATION

UNIQUEMENT sur un mini-projet technique (à 2 ou 3)!

- 2013/2014 : projet « myCloud »
- 2014/2015 : projet (pimp my P2P))
- 2015/2016 : projet ((Lord of the sensors))
- 2016/2017 : projet « Lord of the sensors (le retour) » : one place to rule them all
- 2017/2018 : projet « CMS Check My Sensors »
- 2018/2019 : projet « MySQL My Smart Quiet Locality »
- 2019/2020 : projet « TISSU Tiny Smart Sensors Unified »
- 2020/2021 : projet « SEW Sensors EveryWhere »
- 2021/2022: projet « ID-PHYCS Interactive Disseminated PHYsical Complex Sensors »
- 2022/2023 : projet « Phi-X/it PHYsically eXtended Interactive Technologies »

Sujet donné mi octobre, à livrer mi-janvier (mini-rapport + code + démonstration)



SUIVI



https://github.com/truillet/upssitech/tree/master/SRI/3A/ID

