Réseaux

INSTITUT NATIONAL DES SCIENCES APPLIQUEES DE LYON

FORMATION

Année scolaire 2012 - 2013

GI

Auteur :
Paul Ferrand

membre de
UNIVERSITÉ DE LYON

INSA LYON

Réseaux

Cours magistral

Paul Ferrand

INSA Lyon

Année scolaire 2012-2013

Présentation du cours

Intervenants :

- Paul Ferrand (Laboratoire CITI - Batiment Télécom) : paul.ferrand@insa-lyon.fr
- Christine Michel (Batiment Léonard de Vinci) : christine.michel@insa-lyon.fr

Planning :

- 4 séances de cours magistraux (8h)
- 3 séances de TP (12h)
- 3 séances de projet (12h)
- Interventions extérieures (2h + 4h)

Examen en fin de semestre, et (petite) annexe à votre rapport de stage.

Synopsis du cours

Pourquoi un réseau ?

- Echanger des informations, supporter et accélérer la communication
- Partager ou vendre des services
- Mutualiser des ressources

Pourquoi un cours de réseau en GI ?

- Vous êtes au contact des réseaux dans votre vie personnelle et professionnelle, et vous devez en être des utilisateurs *éclairés*
- Vous serez amené en tant que managers à suivre des projets incluant des réseaux informatiques
- Les systèmes d'information (et donc les réseaux) sont le support du métier de toutes les entreprises
- En tant qu'ingénieurs, vous devez étoffer votre culture et votre curiosité scientifique et technique

Synopsis du cours

Dans son ensemble :

- Comprendre le fonctionnement général des réseaux informatiques
- Connaitre le déroulement d'un projet d'audit ou de conception d'architecture de réseau
- Savoir concevoir et analyser une architecture de réseau simple et d'une infrastructure de système d'information
- Comprendre le fonctionnement d'une infrastructure de sécurité, et savoir l'utiliser quand elle est disponible
- Connaitre les dangers numériques actuels, et les impacts en terme d'intelligence économique
- Connaitre les différentes problématiques juridiques liées aux systèmes d'information et aux réseaux

Historique

Internet, ça marche

- Mais pourquoi ?
- Simple, intuitif (au moins pour votre/notre génération...)
- Des millions de pages et de services, pour des millions d'utilisateurs simultanés
- Mais en fait, l'Internet, c'est quoi ?

Une technologie jeune

- Le World Wide Web (WWW) n'a que 20 ans
- L'Internet « mondial » en a 25
- En quelques années, Internet a changé la société et s'est rendu presque indispensable
- Et son histoire commence il y a à peine 50 ans

1950-1960

Les « ordinateurs » de l'époque étaient énormes, isolés

- Effectuent un seul calcul à la fois (avec des cartes perforées !)
- Accès séquentiel à l'ordinateur, géré souvent par un humain (qui du coup donnait les cartes à manger à l'ordinateur..)
- Développement très long, très peu pratique !
- L'avènement des terminaux et des claviers n'a que peu simplifié les problèmes d'accès

Première « mise en réseau »

- Déporter l'écran de contrôle de la machine, et s'autoriser un accès à distance à l'ordinateur
- Rajouter plusieurs écrans de contrôles pour le même centre de calcul
- Problème de gestion d'accès concurrents, d'utilisateurs, ...

Projets fondateurs

En 1957, lancement de Spoutnik par l'URSS. Les États-Unis ont peur de perdre leur supériorité scientifique, et lancent donc le DARPA (Defense Advanced Research Projects Agency).

- Gérer et financer les projets de recherches
- Premier besoin identifié : faciliter la communication entre les différentes agences de recherche du pays, en particulier les différents centres de calcul
- Création de l'ARPANET en 1966, reliant 3 des plus grandes universités des États-Unis entre elles par des liens permanents

Simultanément

- Réseau militaire de la RAND Corporation
- Réseau commercial du NPL (National Physical Laboratory) en Angleterre
- Réseau de l'INRIA en France, nommé CYCLADES

Projets fondateurs

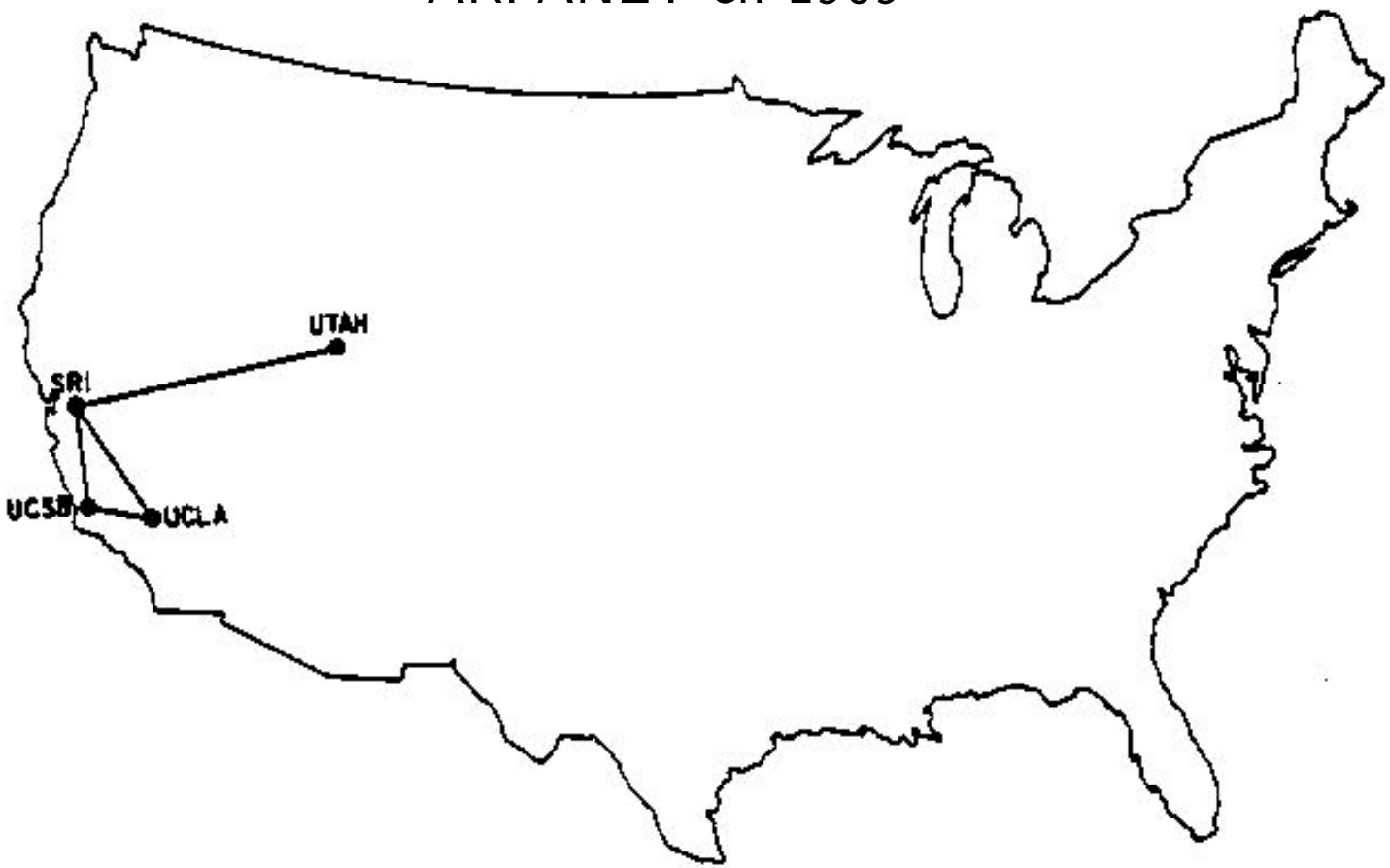
Ces quelques grands projets de l'époque avait des buts différents, et ont tous contribué à la forme de l'Internet actuel par la réponse technologique apportée à leurs besoins.

ARPANET

- Le but premier était de relier entre eux de gros supercalculateurs, et donc de créer un **backbone**
- Les terminaux distants pouvaient donc se trouver géographiquement très loin
- Rapidement, les supercalculateurs ne géraient plus le réseau, cette partie étant prise en charge par des équipements dédiés
- Focalisation sur la fiabilité du transfert de données, et naissance du protocole **TCP** encore utilisé aujourd'hui

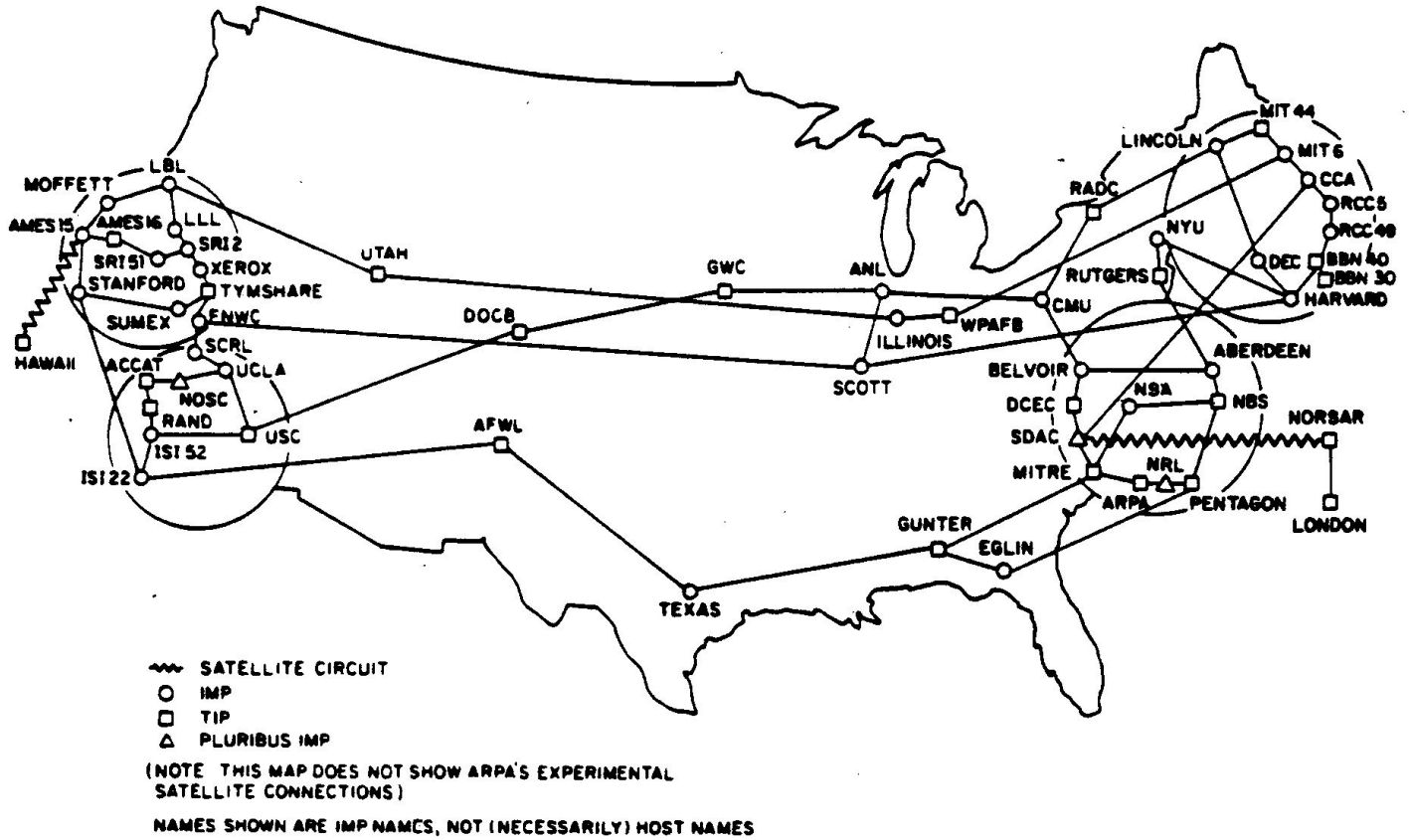
Projets fondateurs

ARPANET en 1969



Projets fondateurs

ARPANET en 1977



Projets fondateurs

RAND

- Réseau militaire sans fils
- Privilégier un grand nombre de relais plutôt qu'une connexion unique à longue distance
- Axé sur la résilience ; perdre un noeud du réseau n'impacte que peu la connectivité globale

NPL

- Basé sur les travaux de l'ARPANET
- Fragmentation des données et commutation de paquet (voir 2nd cours)
- Simplifie la gestion du réseau et des lignes
- Evite les collision et permet à plus d'utilisateurs de partager les accès

Projets fondateurs

CYCLADES

- Fonctionnement fondamentalement différent d'ARPANET
  - Petits réseaux gérés individuellement contre grand réseau central
- Routage distribué
  - « Inter-networking »
  - Relier des petits réseaux entre eux par un maillage dense
  - Comme le RAND, on pouvait trouver plusieurs routes entre deux points
- Noeuds intermédiaires stupides
  - Les équipements terminaux, aux extrémités, sont responsable de la transmission
  - Les noeuds intermédiaires ne font que relayer
  - Structure de bout-en-bout
- L'intelligence est donc reléguée à l'extrémité du réseau, et non au centre, ce qui est à l'origine de la grande robustesse d'Internet aujourd'hui

Unification et expansion

- L'ISO (International Standard Organization) tente de fédérer tout ce beau monde dans les années 1980, sans grand succès
- L'ARPANET intègre tout de même certaines recommandations de l'ISO, ainsi que les avancées des projets similaires, et forment la pile « TCP-IP » telle qu'on la connait aujourd'hui (plus de détails la semaine prochaine)
- L'ARPANET s'étend et devient « NSFNet », les premiers liens satellites transatlantiques apparaissent dans les années 1980
- Dans les années 1990, les liens transpacifiques se multiplient (Singapour 1990, Japon 1992, Australie 1994...)
- NSFNet est décommissioné en 1995, les backbones deviennent commerciaux

Internet : réseau des réseaux

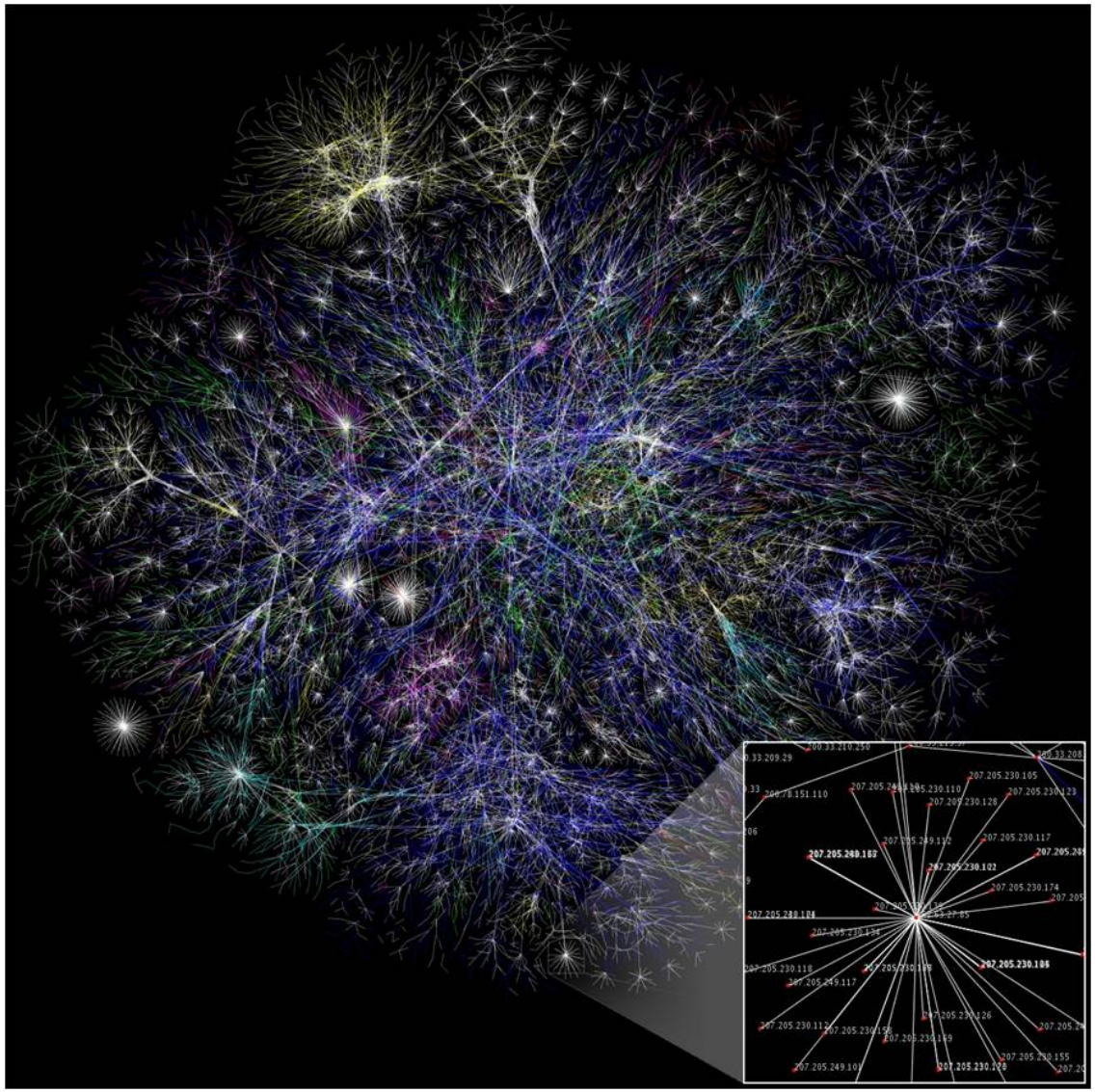
Du coup, sans ARPANET, qui contrôle Internet aujourd'hui ?

- En fait, Internet « n'appartient » pas à une seule entité
- C'est une agrégation de réseaux de différents prestataires qui se sont mis d'accord pour **s'interconnecter**
  - Par exemple, en France, France Telecom et Illiad (maison mère de Free) possèdent un réseau conséquent
  - Ils vont passer des accords commerciaux avec d'autres réseaux pour échanger de l'information
- A l'heure actuelle, un réseau solitaire n'a quasiment aucune valeur commerciale, les opérateurs sont donc forcés de payer ces interconnexions pour leurs utilisateurs
- **On a donc une première vision d'Internet comme un ensemble de petits et grands réseaux reliés entre eux**

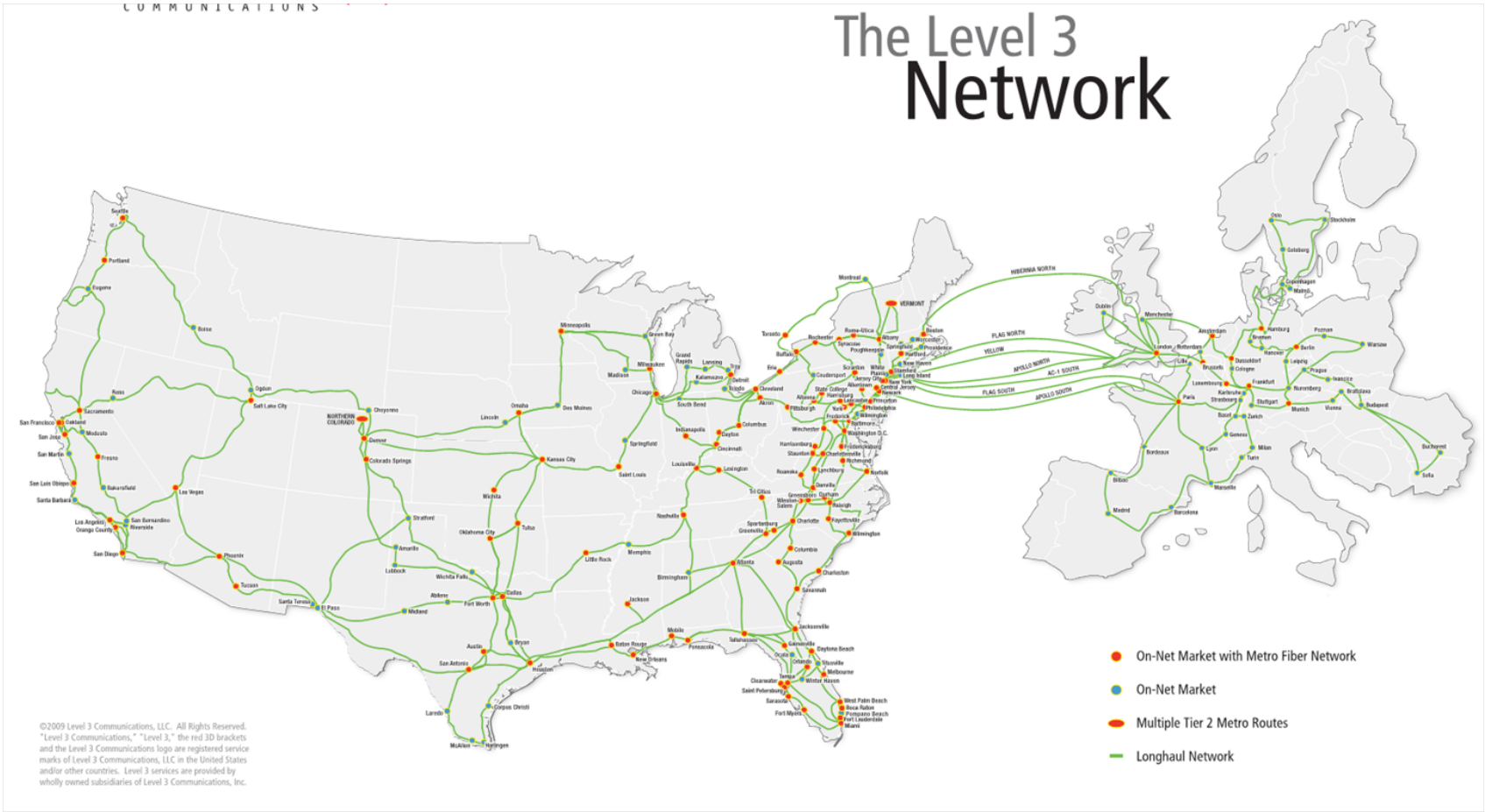
Internet : réseau des réseaux

- Certaines entreprises possèdent des réseaux trans-continentaux (les **backbones**)
  - On les appelle les opérateurs **Tier 1**, ils sont environs une douzaine
  - Ces opérateurs ont des contrats d'échange entre eux pour la plupart, et vendent leurs services aux opérateurs **Tier 2**
- Les opérateurs en dessous (hiérarchiquement), les fournisseurs d'accès internet, les fournisseurs de contenus (Youtube, Netflix, Google, ...) sont tous liés aux opérateurs Tier 1 par le biais des **IXP (Internet Exchange Points**
  - On appelle cette opération le **peering**
  - L'accès aux IXP est régulé commercialement entre les partenaires présents
- Héritage de CYCLADES
  - Vous n'êtes pas au courant de ces IXP, car vous vivez à l'extrémité
  - L'augmentation des peerings et des IXP rend le réseau **fortement maillé** et complexe
  - Plusieurs routes existent pour une même destination : résilience

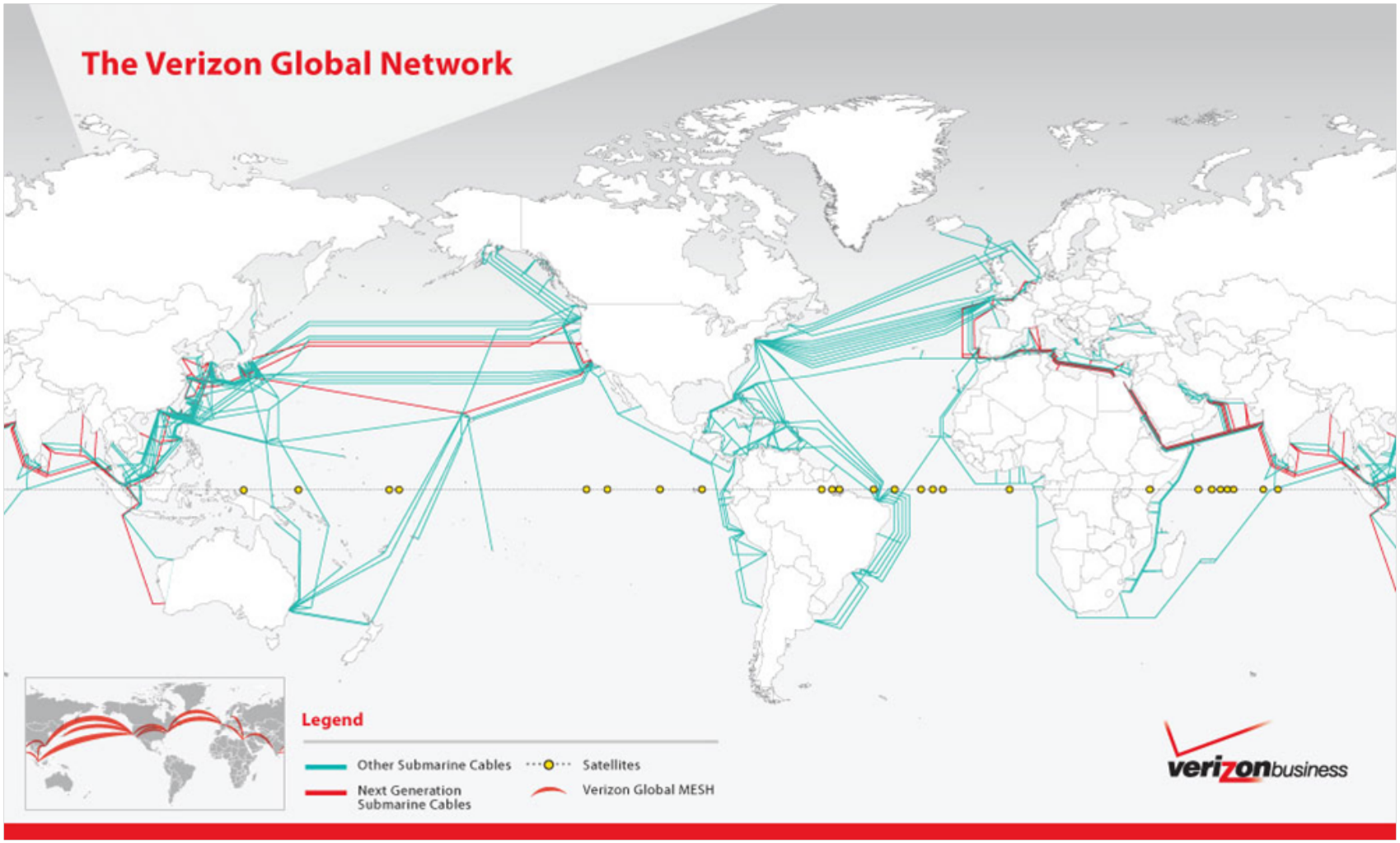
Internet : réseau des réseaux



Internet : réseau des réseaux



Internet : réseau des réseaux



Internet : réseau des réseaux

- Trajet réseau avec `traceroute` de l'INSA vers Wikipédia :

```
 1  psrl146.univ-lyon1.fr
 2  te1-2-cisrezo222.rocad.fr
 3  te5-5-rtr-doua2.rocad.fr
 4  193.55.215.222
 5  * * *
 6  te0-3-0-0-lyon1-rtr-001.noc.renater.fr
 7  te1-1-lyon2-rtr-021.noc.renater.fr
 8  te0-0-0-1-paris2-rtr-001.noc.renater.fr
 9  typhon.franceix.net
10  10ge-6k3.6k-2.th2.fr.typhon.net
11  20ge-6k2.th3.fr.typhon.net
```

- Passage de **ROCAD** (Réseau de la Doua) vers **RENATER** (Réseau de l'Enseignement Supérieur) vers **FranceIX** puis **Typhon**, fournisseur de Wikipédia !

Internet : support des communications

E-mail

- Une des premières utilisations des réseaux (répond à un besoin de communication fondamental)
- 1965 : commande `mail` des systèmes UNIX
- 1971 : premier mail envoyé sur l'ARPANET à travers le réseau
- 1973 : le mail est déjà, et restera, la principale utilisation du réseau

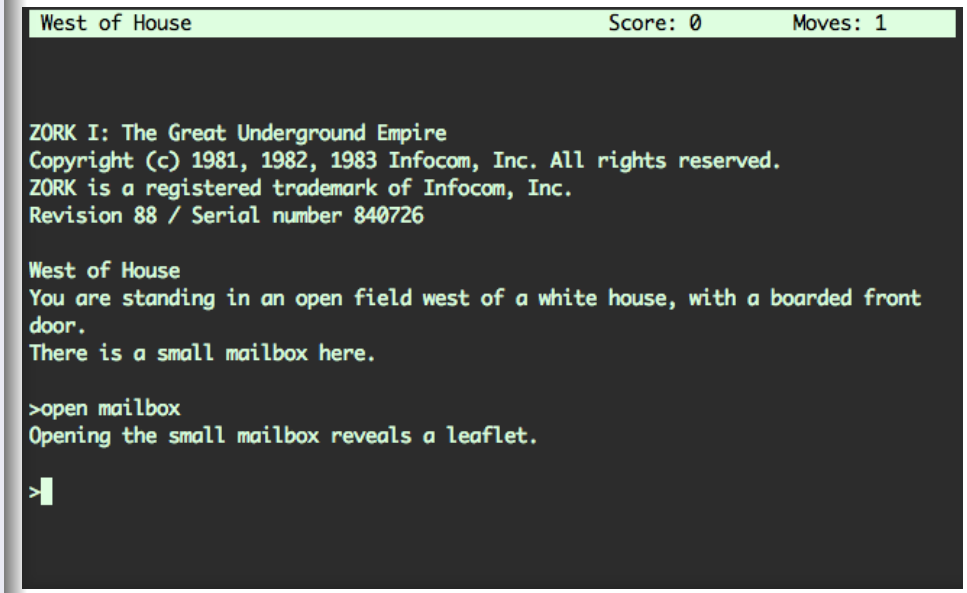
Forums

- 1978 : les BBS (bulletin board systems)
  - Peu conviviaux, permettent uniquement de laisser des messages dans un « tas » non trié
  - Mais première idée de lieu d'échange ouvert à tous
- 1979 : Newsgroup et Usenet
  - Organisation des messages sous forme de sujet de discussion
  - Regroupement des sujets par centre d'intérêt, appelés des hiérarchies

Internet : support des communications

Les jeux en ligne

- Toutes ces technologies sont avant tout utilisées par des « purs » informaticiens
- 1979 : Premier jeu de rôle coopératif en ligne
- En mode texte et sans images
  - Les jeux en image en ligne sont arrivés bien après
- World of Warcraft n'a rien inventé !



Internet : support des communications

Le smiley

- Les smileys (plus généralement *emoticons*) ont été rempants depuis le début du 20e siècle.
- Dans les télécommunications, une des premières apparitions d'une suite de caractères pour représenter une émotion a été proposée en 1979.
- Les yeux sont apparus en 1982

```
I propose that the following character sequence
for joke markers:
:-)

Read it sideways.  Actually, it is probably
more economical to mark things that are NOT jokes,
given current trends.  For this, use :-(
```

Internet : support des communications

Le chat

- Encore une fois, invention Française (même si ya pas forcément de quoi être fier..)
- Le minitel : utilisé entre autre pour parler en ligne, simplement à partir d'une ligne téléphonique, à ses connaissances
- Mais aussi, parler à des gens que l'on ne connait pas du tout
  - Premiers sites de rencontres
  - Premières lignes roses

Internet : support des communications

Le World Wide Web (WWW)

- Inventé en **1990** par un chercheur du CERN
  - Mettre en ligne, en accès libre, des documents textuels reliés entre eux par des liens **hypertextuels**
  - Des chercheurs avaient néanmoins posé le concept dans les années 1950
- Protocole **HTTP**, que l'on détaillera plus en avant de ce cours
- Naissance de l'Internet tel que pratiqué aujourd'hui
- 1995 : premières pages web **personnelles** sur Geocities, et démocratisation des accès Internet
  - Internet changea d'interface, et devint un joyeux bordel...
- Les gens veulent trier ces informations
  - Naissance des **moteurs de recherche** et des **weblogs**

Internet : support des communications

Wikipedia

- Une idée stupide : « N'importe qui peut écrire dans mon site »
- En 10 ans, Wikipédia a écrasé les encyclopédies historiques, et est devenu la référence instantanée de beaucoup de personnes
- La controverse est toujours forte, mais le système tend à se réguler par lui-même

Réseaux sociaux

- Immense phénomène de société, très récent !
  - 1997 pour *SixDegrees.com*
  - 2003 pour LinkedIn
  - 2006 (!) pour Facebook et Twitter
- Avancée logique, à posteriori, du monde communicant poussé à l'extrême
- Près d'un milliard d'utilisateurs sur Facebook et Twitter

Internet : support des communications

- **Seconde vision d'Internet, comme une plate-forme de services**
- Bulle internet des années 2000
  - Plusieurs entreprises ont tenté de creuser leur trou, en répondant ou en identifiant des utilisations potentielles d'Internet
- Le concept d'infrastructure de services est pervasif pour les métiers de l'entreprise
  - Buzzwords : SOA (service-oriented architecture), B2B/B2C (Business to Business/Consumer), Webservices, Cloud Computing, ...
  - Change la manière dont sont conçus et maniés les systèmes d'information dans les entreprises
  - **Réorganisation en services** de l'informatique, externalisation, ...
- Les services offerts sur Internet évoluent encore rapidement aujourd'hui
  - Vidéo à la demande, téléphonie, Deezer/Spotify ...
  - Changement des modes de consommation et des attentes des utilisateurs

Les petits malheurs des ingénieurs télécom

Transmission physique

- Différents modes et **médiums de transmission**
  - Plus d'informations au prochain cours
  - Mais vous les connaissez déjà, principalement des câbles ou des ondes !
- Pour chaque médium de transmission, il faut physiquement un moyen de se comprendre
  - Modèle **numérique**, suite de 0 et de 1
  - Comment manipuler les phénomènes physiques pour **transformer une information numérique en une information analogique**
- Grande variété de modes de transmissions et d'équipements, qui évoluent très rapidement
  - Nécessité de standardiser pour l'utilisateur, transparence du médium de transmission
  - L'utilisation du réseau reste la même si l'on est câblé ou en Wifi...

Les petits malheurs des ingénieurs télécom

Collisions et contrôle d'accès

- Partage du médium de transmission (par exemple sur le Wifi)
- Si tout le monde parle en même temps ?
  - Perte d'information, interférence et réduction de performances
  - Définir qui parle, quand, combien de temps...
- Notion de « contrôle d'accès au médium », en anglais MAC

Détection d'erreur et correction

- Le monde physique n'est pas parfait, et les médiums de transmission induisent des erreurs
  - Soit on peut vivre avec (téléphone, télévision, ...)
  - Soit on doit les détecter et demander un renvoi
  - Ou bien trouver un moyen technologique de les corriger ?
- Doit-on corriger au niveau du médium de transmission à chaque saut, ou de bout-en-bout ?

Les petits malheurs des ingénieurs télécom

Interprétation des données

- Imaginons que mes données aient été transmises, corrigées, etc
- Qu'est-ce que je reçois ?
  - Un mail ? Une musique ? Une image ?
  - Comment est codée mon information ? Dans quelle langue ? Combien d'images/seconde pour ma vidéo ? ...

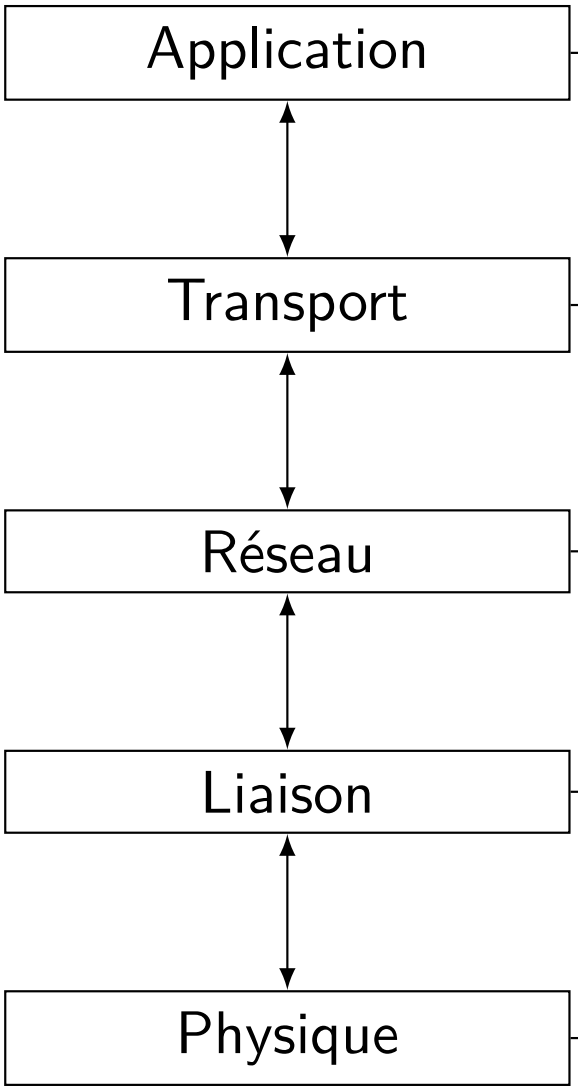
Contrôle d'accès et authentification

- Problématiques de sécurité
- Comment s'assurer que personne n'espionne ma transmission ?
- Comment faire pour connaitre l'identité de mon interlocuteur ?
- Si l'on m'écoute, puis-je cacher le message que j'envoie de manière fiable ?

Modèle en couches

- Ces problèmes sont classiques, et ont été rencontrés puis résolus par les ingénieurs et les chercheurs en informatique et réseaux
- Deux notions importantes :
  - **Diviser pour mieux régner** ; il est plus simple de traiter chaque problème séparément
  - Accord entre les interlocuteurs et notion de **protocole**
- Principe du modèle en couche
  - Chaque couche a une fonction précise qui correspond à un problème à résoudre
  - Chaque couche communique à la couche directement supérieure et inférieure **uniquement**
- On retrouve les deux visions du réseau
  - Vue « technique » où le modèle en couche va permettre de fournir les réponses technologiques pour communiquer
  - Vue « service » ou l'utilisateur communique avec la partie supérieure du modèle en couche et utiliser le réseau

Modèle en couches



Lien entre l'utilisateur et le réseau. **Siège des applications** que vous connaissez.

Contrôle de la communication de bout-en-bout. Assure la **délivrance** et le **contrôle des erreurs**.

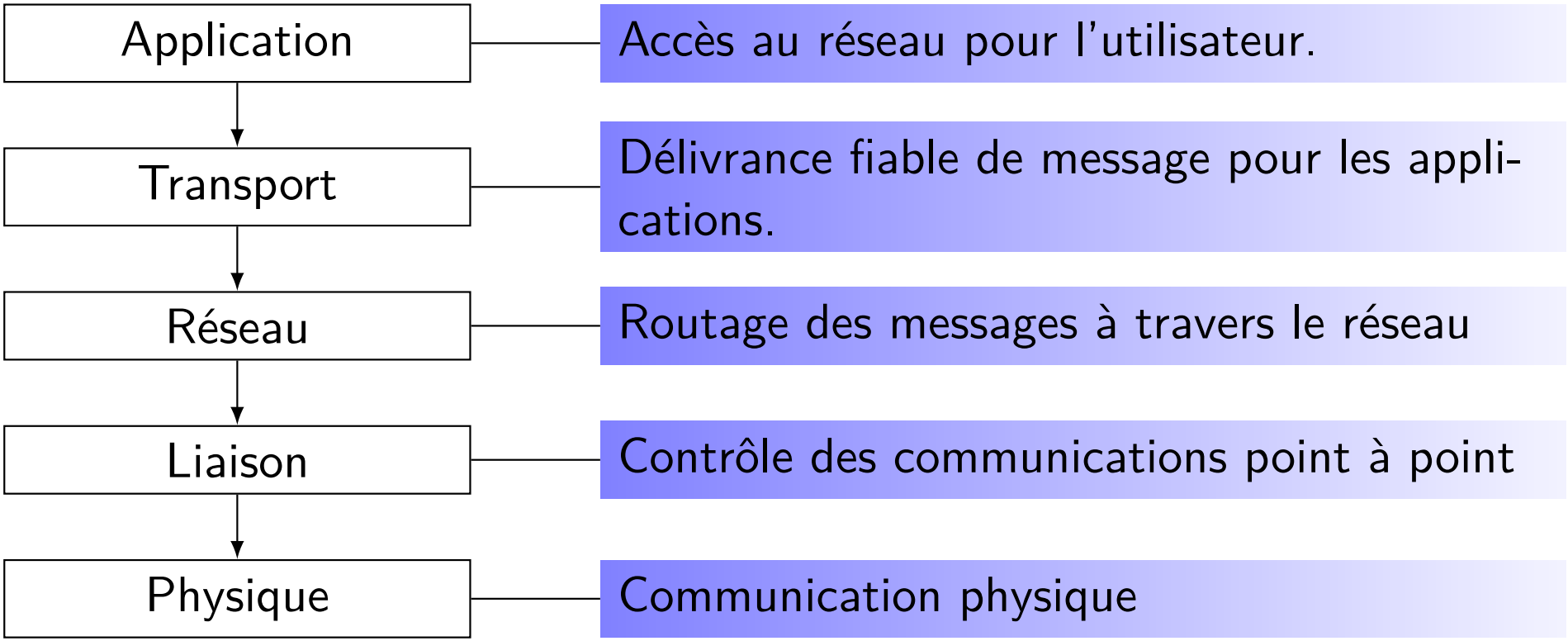
Gestion de l'acheminement dans le réseau, **adressage** et choix des routes.

Contrôle de l'accès au médium de transmission et détection d'erreur sur les liens inter-équipements.

Interfaces de connection, et **transmission physique** de l'information numérique.

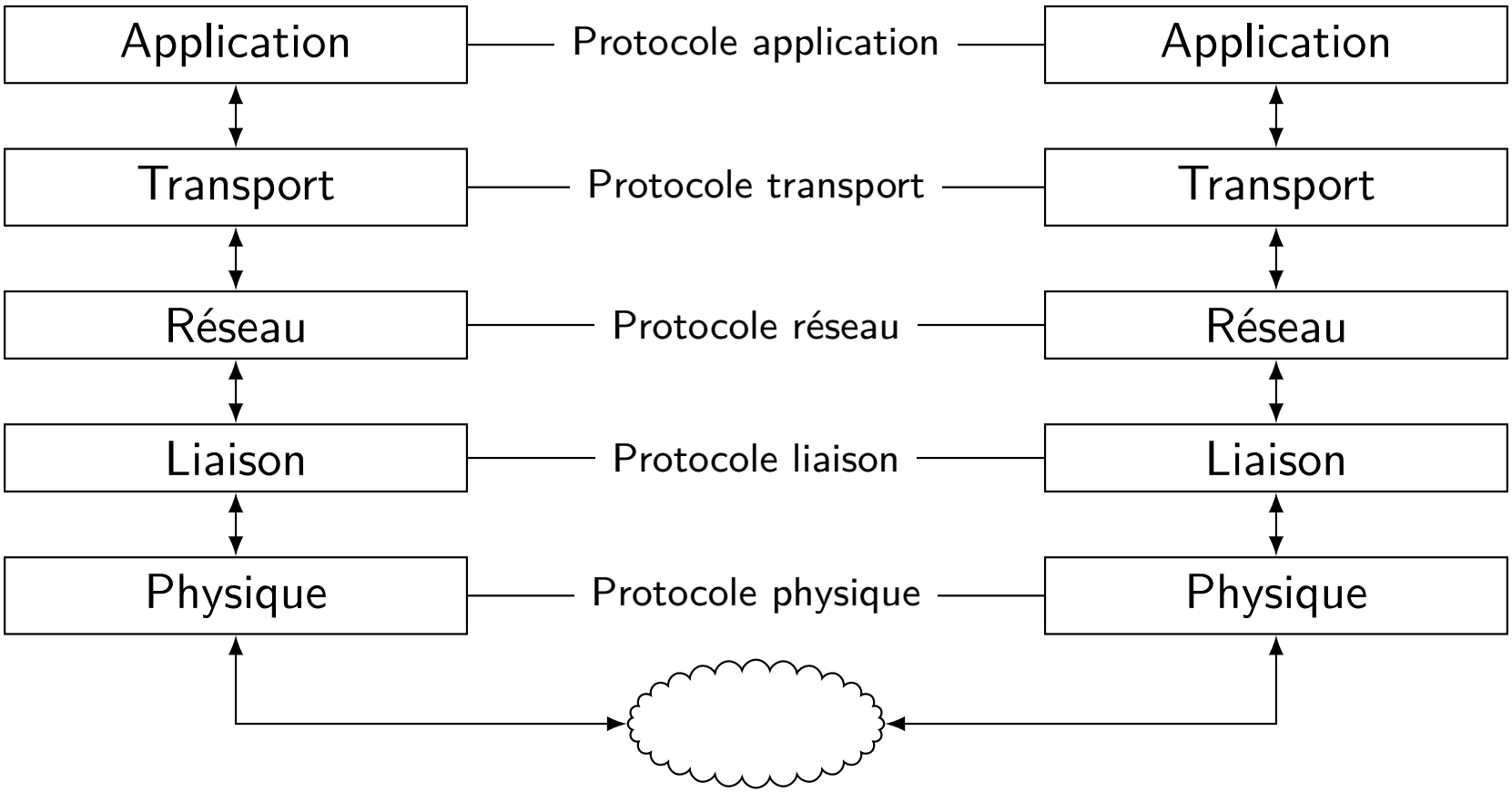
Modèle en couches

Vision service



Notion de protocole

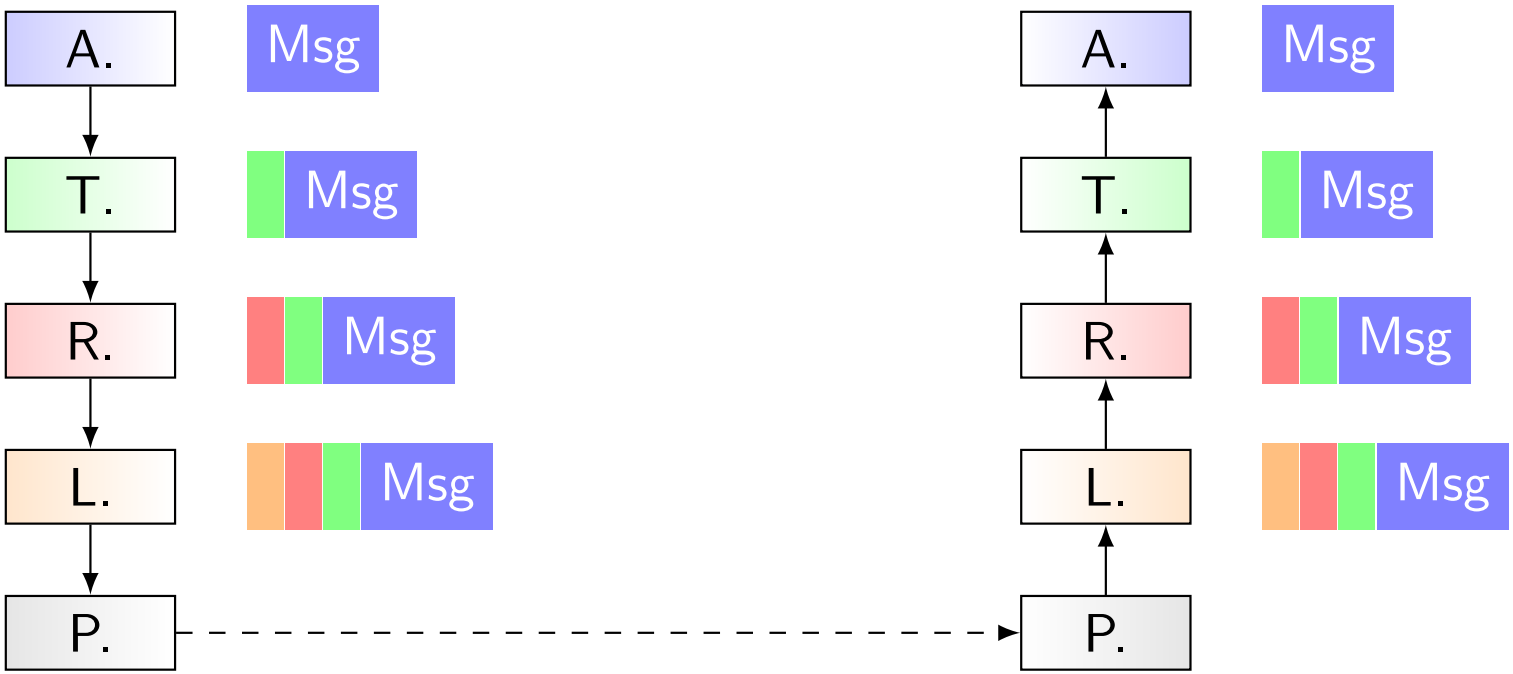
La notion de protocole est juste un certain nombre de règles qui régissent la communication.



Modèle en couches

Encapsulation

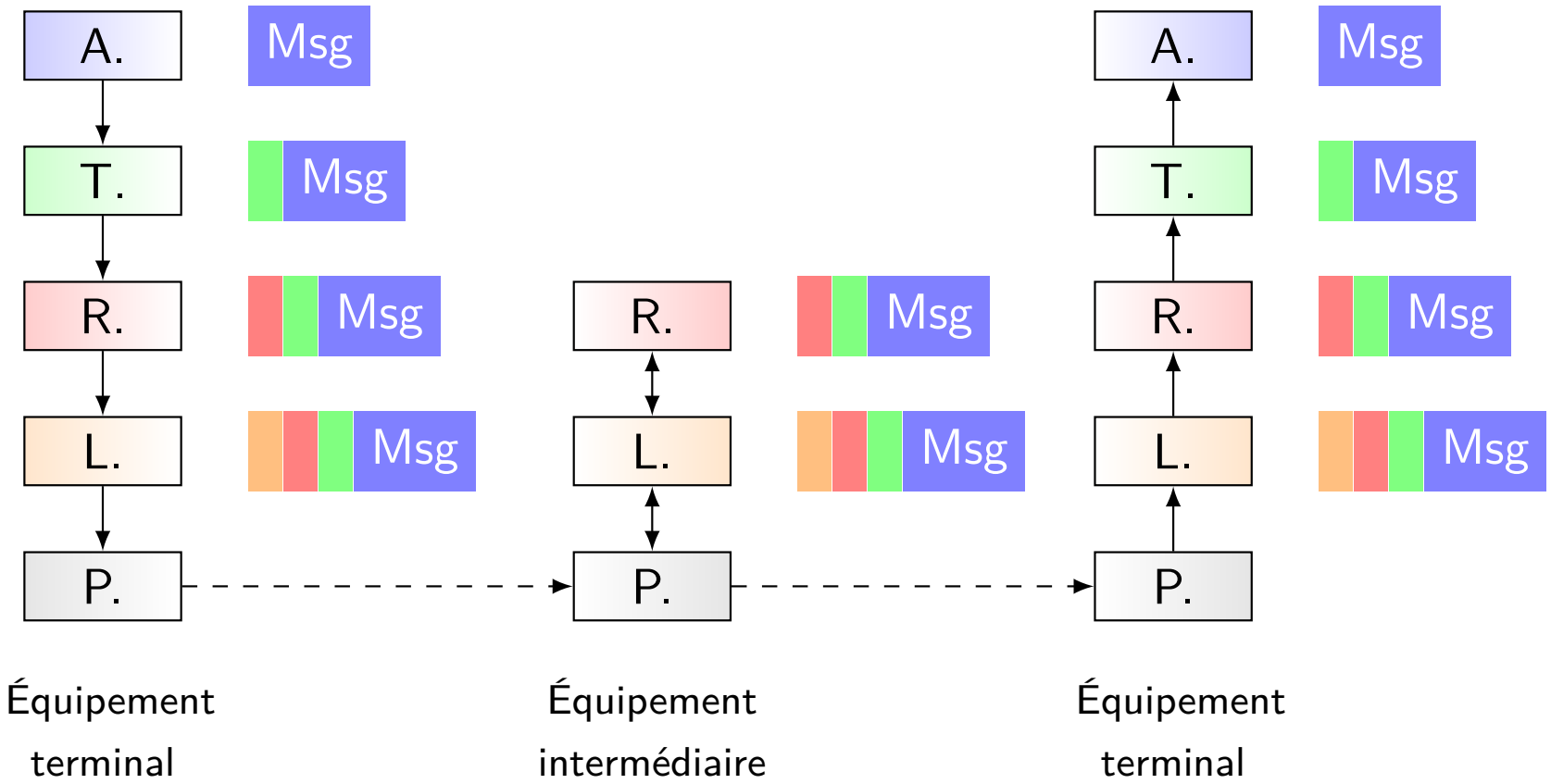
- Les protocoles communiquent entre eux uniquement, pour transmettre les données des couches supérieures
- Utilisation **d'entêtes** pour communiquer entre protocoles, sans altérer le contenu du message



Modèle en couches

Encapsulation

- Les noeuds intermédiaires ne décodent pas toute l'information
  - Simplification de la conception des équipement réseau : à chacun son travail !



Couche application

1. Un protocole de la couche Application tourne sur des **équipements terminaux**, et communique sur Internet
   - Par exemple un navigateur Web communique avec un serveur Web
   - *Ne s'intéresse pas aux équipements de coeur de réseau*
2. Aspects de l'implémentation de protocoles de la couche Application
   - Modèles de service de la couche transport
   - Paradigmes **client-serveur** et **peer-to-peer**
3. Étudier des protocoles courants : HTTP et SMTP (en TP)
   - Comprendre leur fonctionnement et leurs limitations
   - Savoir analyser rapidement des traces réseau
4. Découvrir la programmation réseau (en TP)

Couche application

Deux grands types d'architecture applicative sur les réseaux :

Client-serveur

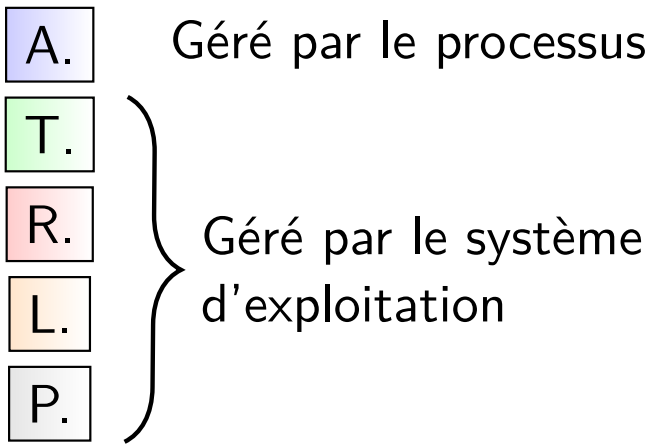
- Le **serveur** est une application, hébergée par une machine **hôte** toujours allumée, avec un accès permanent et fixe
- Le **client** communique avec le serveur de manière intermittente
- Les clients ne communiquent pas directement entre eux, mais passent par le serveur

Pair à pair (Peer-2-peer)

- Pas de serveur au sens de la définition ci-avant
- Les équipements terminaux communiquent entre eux directement, et s'échangent un **service**
- Performances intéressantes, mais gestion compliquée car les équipements changent constamment

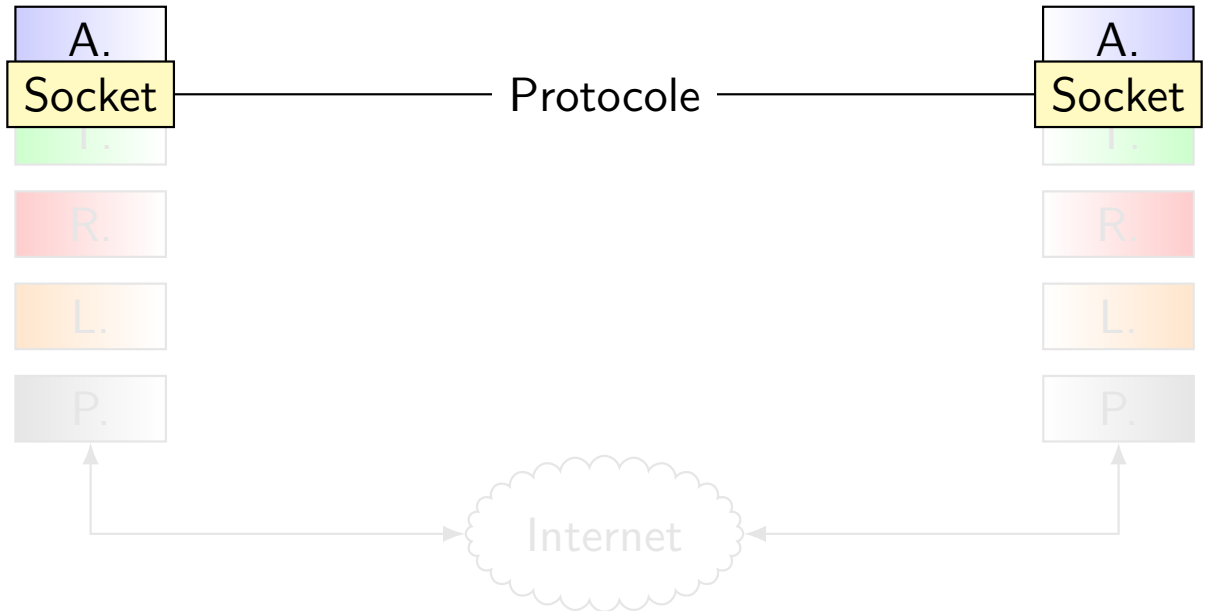
Couche application

- Les applications sont gérées dans des **processus** au niveau du système hôte, et se développent comme tout programme informatique
  - La notion de processus est intégrée dans le système d'exploitation de l'ordinateur
  - Le système d'exploitation offre aux processus la possibilité de communiquer entre eux sur l'hôte
  - Et par le biais de sa **pile de protocole**, la possibilité de communique avec des hôtes différents
- On parle de processus clients et des processus serveurs
  - Les applications pair à pair ont des processus clients et serveurs simultanément



Couche application

- Les processus utilisent le concept de **sockets** pour communiquer avec des hôtes distants, par la couche Transport
  - Le *service* de la couche Transport est donc accessible à la couche application par l'intermédiaire des sockets
  - Le processus envoie des données dans le socket, et récupère les réponses par le socket
  - Le système d'exploitation informe l'application que des données sont disponibles pour elle



Couche application

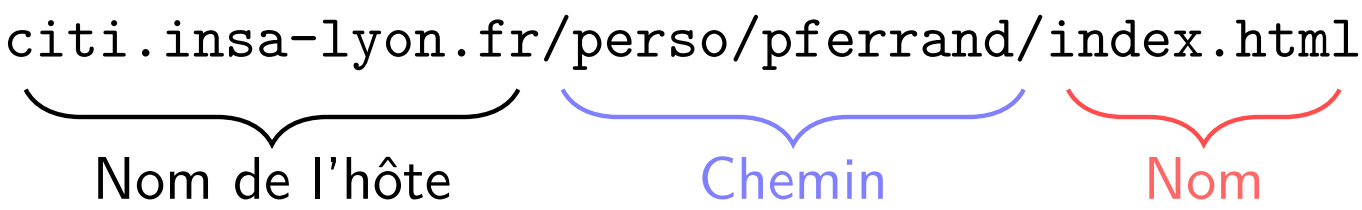
- Le processus doit posséder un identifiant pour recevoir les messages d'autres processus
  - L'adresse IP identifie l'hôte (Cours 2)
  - Mais plusieurs processus peuvent s'exécuter en parallèle sur un hôte
- On utilise donc un second artefact pour identifier les sockets, et les processus qui y sont liés : le **numéro de port**
- Pour communiquer avec une application distante, il faut donc connaitre son adresse IP, et son numéro de port
- Pour les services les plus courant, les numéros de port sont standardisés :
  - Serveur HTTP : port 80
  - Serveur SMTP : port 25
- Les **sockets clients** possèdent aussi un port pour les identifier, tiré aléatoirement par le système d'exploitation, entre 20 000 et 65536

Couche application

- Les applications attendent un certain niveau de service de la part de la couche transport
  - Fiabilité des données et des transferts
  - Arrivée en « temps-réel »
  - Débit minimal
  - Sécurité des données
- Il existe deux standard pour les couches transport :
  - Le protocole **TCP** (Transmission Control Protocol), assure la **fiabilité des données** et le **contrôle de flux et de congestion**
  - Le protocole **UDP** n'assure rien du tout...
- Exemple d'applications, réfléchissez à leurs besoins par rapport à la couche transport !
  - Le transfert de fichier
  - La téléphonie
  - Les jeux en ligne
  - Les e-mails

Le protocole HTTP

- Utilisation de la notion d'hypertexte, où un document contient des **liens** vers d'autres objets ou documents
- Une **page web** est formée d'un fichier écrit en langage HTML (Hypertext Markup Language), qui référence plusieurs objets
- Chaque objet est identifié par une **URL** (Universal Resource Locator) :



- Les objets sont stockés sur un serveur HTTP (appelé couramment serveur Web)
- Même si d'autres langages sont utilisés pour gérer l'application sur le serveur, les pages et la syntaxes suivent les standards HTML et HTTP

Le protocole HTTP

- Comme beaucoup de protocoles, le principe d'HTTP est très simple
  - Modèle client-serveur
  - Le client se connecte et demande une page au serveur
  - Le serveur lui répond avec la page demandée, ou un message d'erreur
- Le protocole fonctionne sur **TCP** et utilise le port 80
- Par défaut, HTTP est dit « sans-état »
  - Les serveurs ne gardent aucune information sur les clients
  - Une fois la requête terminée, la connexion est perdue
- Le protocole spécifie le format des requêtes et des réponses « standards » que les serveurs et les navigateurs Web doivent implémenter
  - Créer un nouveau navigateur, c'est donc tout simplement savoir implémenter les 2 requêtes de bases du protocole HTTP !
  - En fait, la difficulté est surtout d'afficher correctement les fichier HTML à l'écran, mais ça n'est pas le but fondamental du protocole

Le protocole HTTP

- Deux requêtes principalements
  - **GET** : pour récupérer un fichier sur le serveur
  - **POST** : pour envoyer des informations au serveur
- Format d'une requête HTTP :

```
GET /index.html HTTP/1.1
Host: www-net.cs.umass.edu
User-Agent: Firefox/3.6.10
Accept: text/html,application/xhtml+xml
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7
Keep-Alive: 115
Connection: keep-alive
```

Le protocole HTTP

- Un chiffre et un message, en particulier le fameux `404 Not Found`
- Format d'une réponse HTTP :

```
HTTP/1.1 200 OK
Date: Sun, 26 Sep 2010 20:09:20 GMT
Server: Apache/2.0.52 (CentOS)
Last-Modified: Tue, 30 Oct 2007 17:00:02 GMT
ETag: "17dc6-a5c-bf716880"
Accept-Ranges: bytes
Content-Length: 2652
Keep-Alive: timeout=10, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=ISO-8859-1
BLOC DE DONNÉES
```

Le protocole HTTP

- Pour les applications modernes, besoin de gérer les **états**, ou de personnaliser les pages web suivant l'utilisateur
- Utilisation d'une base de donnée côté serveur, et gestion de **cookies**
  - Lors de la première connexion, le serveur crée un **identifiant de session** unique et demande au navigateur de l'utilisateur de s'en souvenir
  - Lors des connexions suivantes, le navigateur renvoie le cookie correspondant au site pour que le serveur le reconnaisse

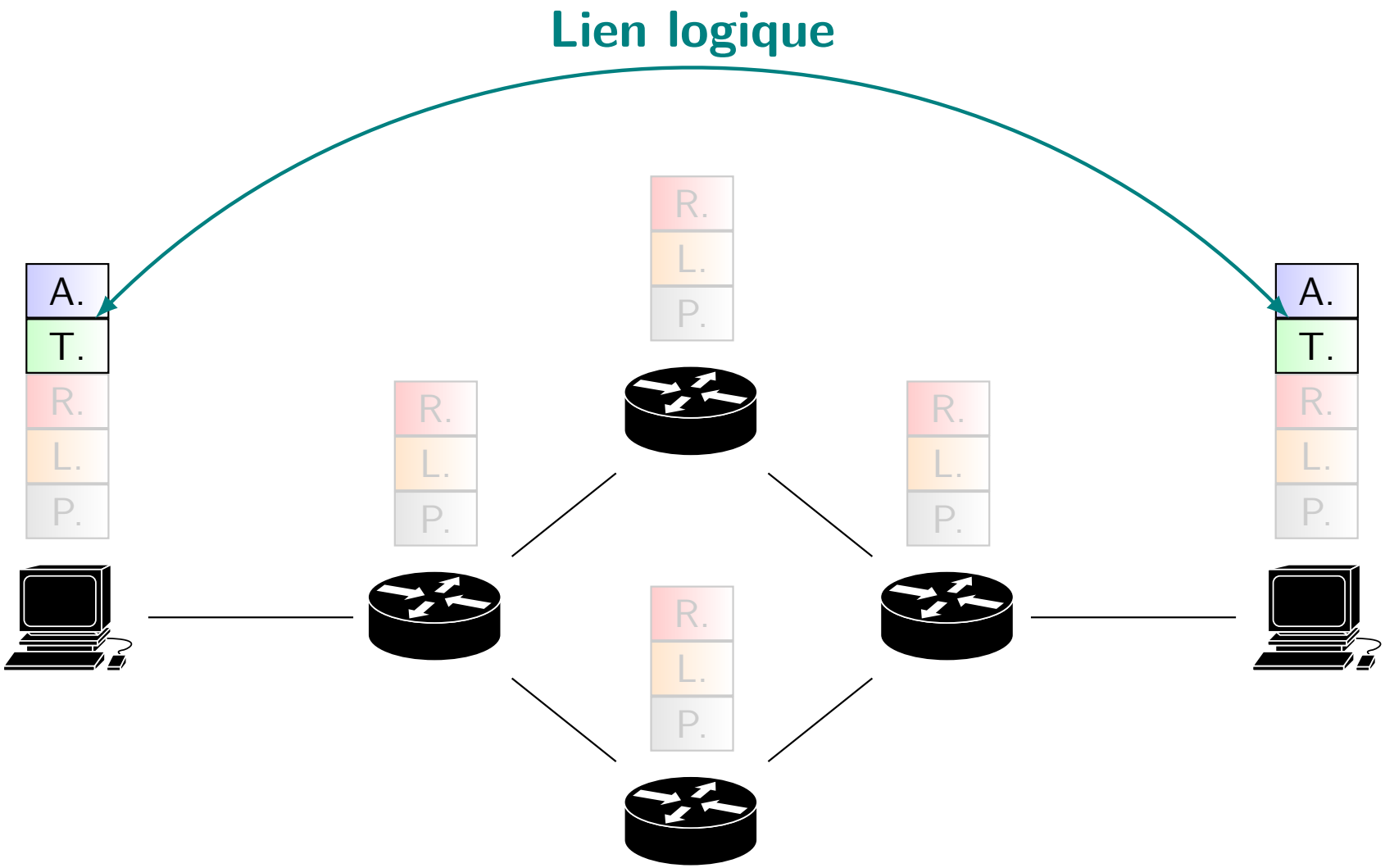
- Les cookies rendent l'utilisation d'internet plus agréables, mais posent des problèmes de sécurité
  - Stockage et identification automatique des utilisateurs, à des fins commerciales
  - Vol de cookies et usurpation d'identité (comme on pourra le voir lors du TP 3)

Couche transport

- La couche transport autorise les communication **entre les applications** d'un réseau
  - Du point de vue de l'application, la connexion réseau est directe et ne passe pas par des équipements intermédiaires
  - Création d'un lien logique entre les processus hébergeant les applications sur des machines différentes

- La couche transport est donc implémentée uniquement sur les équipements terminaux, et assure la communication **de bout en bout**
  - En accord avec les objectifs conceptuels de l'Internet
  - Quelques équipements intermédiaires (firewalls...) peuvent néanmoins « observer » la couche transport

Couche transport

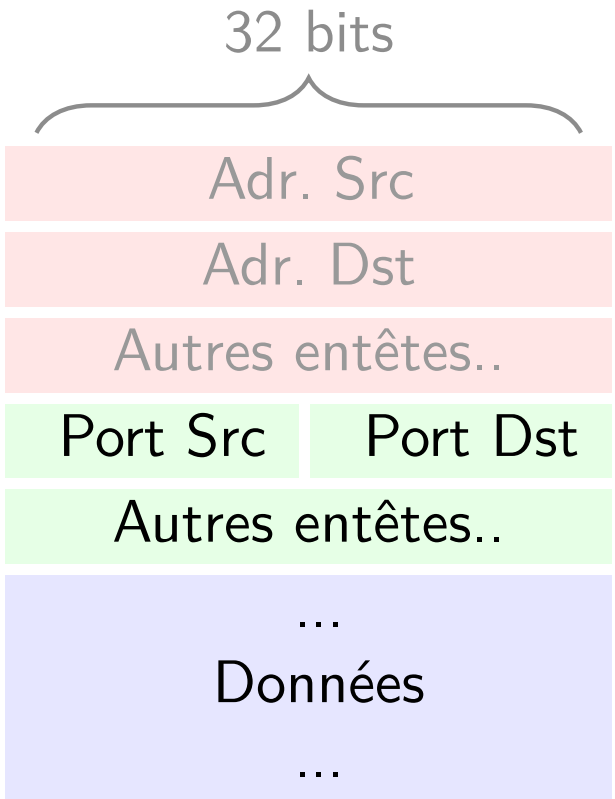


Couche transport

- Différences entre la couche transport et la couche réseau
  - La couche réseau assure la communication **entre les hôtes**, alors que la couche transport l'assure **entre les applications**
  - Analogie avec une adresse postale, et le destinataire précis à une adresse postale
  - La couche réseau d'Internet n'assure qu'un strict minimum de qualité de service, donc les besoins des applications doivent être couvert par la couche transport
- Le premier rôle de la couche transport est le **multiplexage**
  - Pour chaque hôte, plusieurs applications en parallèle
  - Mais les hôtes ont une seule adresse (IP), et l'on utilise donc une second valeur pour identifier le processus sur une machine : **le numéro de port**
  - Une **connexion** (logique) est donc représentée par 4 valeurs : `(Adresse source, port source) - (Adresse dest. port dest.)`

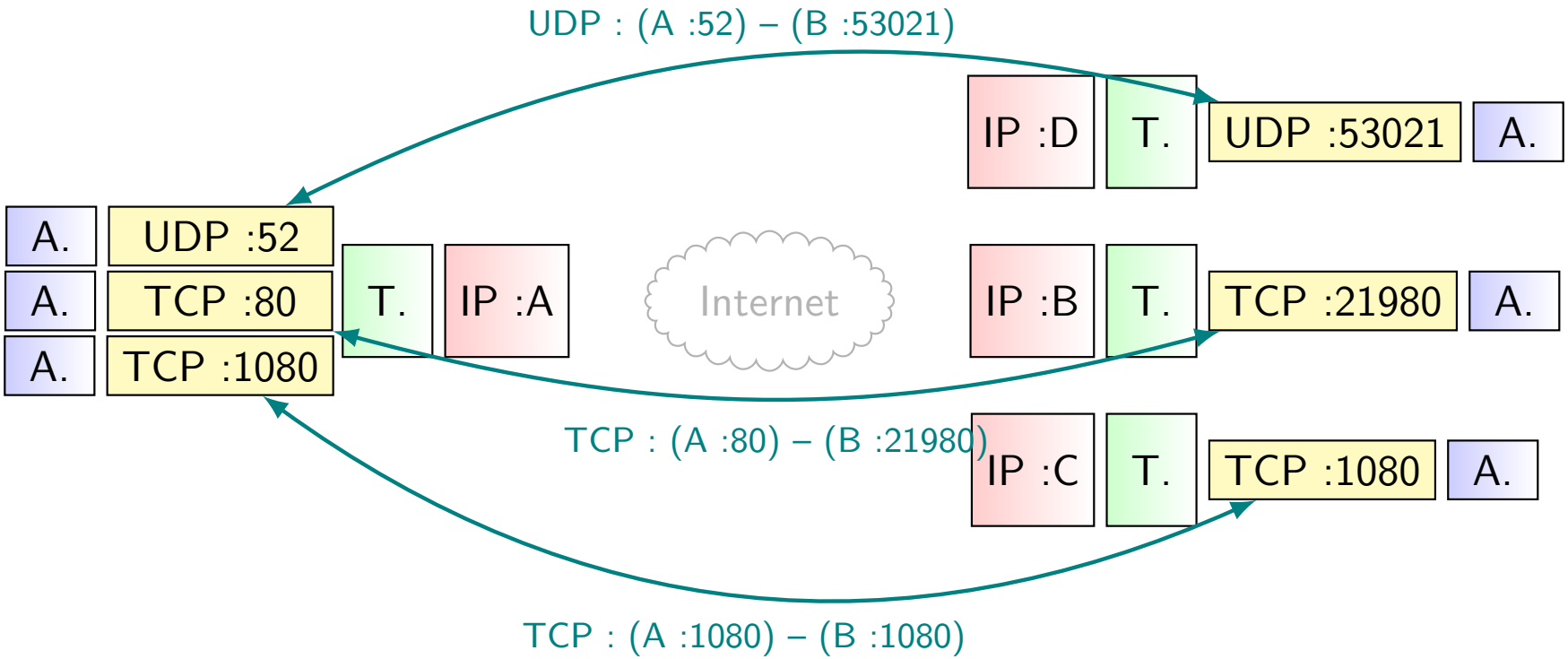
Couche transport

- Passage d'information depuis la couche 3 vers les applications
  - Service réseau offert à travers les **sockets**
  - Le système d'exploitation (OS) gère les sockets et les applications qui y sont liées
  - La couche transport examine les paquets et les dirige vers les bons sockets
- `netstat -bnp TCP` permet de voir les processus et les sockets associés (sous Windows)



Couche transport

- Méthodes de multiplexage des couches transport
  - **UDP** : socket identifié uniquement avec le port de destination
  - **TCP** : socket identifié aver l'adresse source ET les numéros de port
- Notion de socket client et serveur
  - Un serveur HTTP va recevoir toutes ses connexions sur le même port

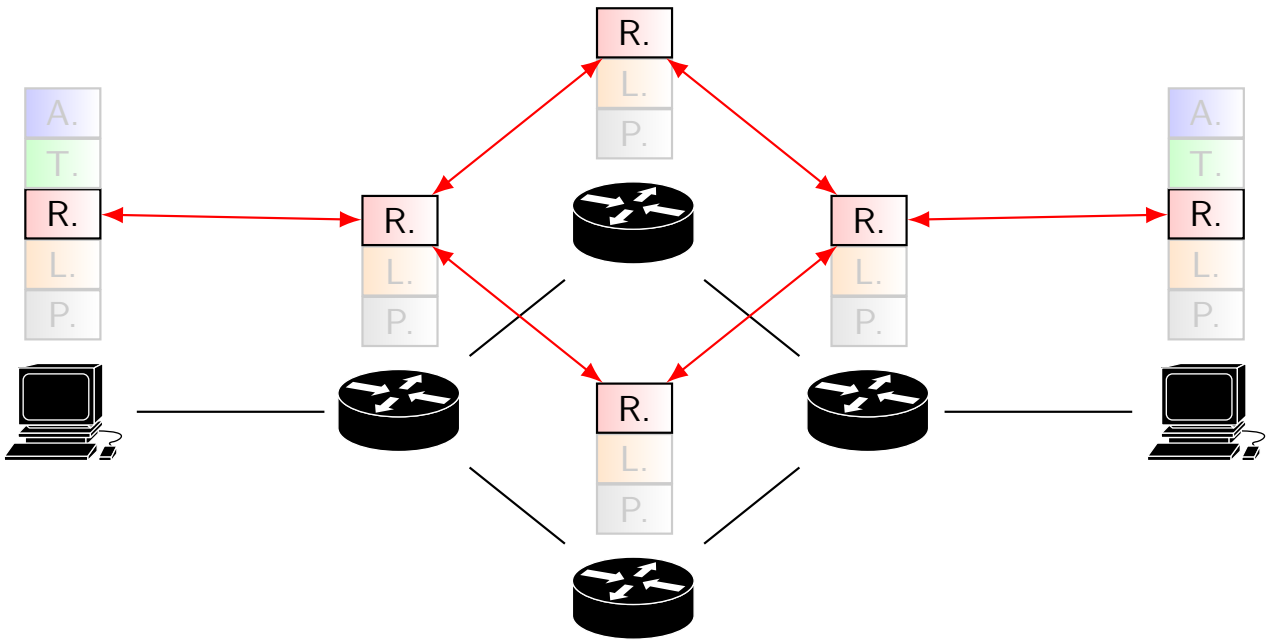


Couche transport

- UDP assure donc l'opération de multiplexage pour la couche transport, et rien d'autre
- TCP possède plus de fonctionnalités
  - **Contrôle des connexions** : il faut « ouvrir » une connexion TCP avant d'envoyer des données, par un **handshake**
  - **Contrôle des données** : les paquets TCP sont numérotés, et la réception doit être confirmée
  - **Contrôle de congestion** : TCP peut détecter une surcharge du réseau et moduler son débit de transmission
  - TCP remet également les paquets dans le bon ordre à la réception, si besoin
- Pourquoi utiliser UDP ?
  - Pour assurer toutes ces fonctions, l'entête de TCP est plus large (20 octets, contre 8 pour UDP)
  - Le contrôle de congestion implique que l'envoi de données peut être retardé
  - De même, le contrôle des données peut retarder la réception

Couche réseau

- Passage de la périphérie du réseau (Application et Transport) vers le **coeur de réseau**
- Le rôle de la couche réseau et l'acheminement des paquets **d'hôte à hôte**
  - Pour la couche transport, c'était de **processus à processus**
- Contrairement à la couche transport, la couche réseau est implémentée dans presque tous les équipements !

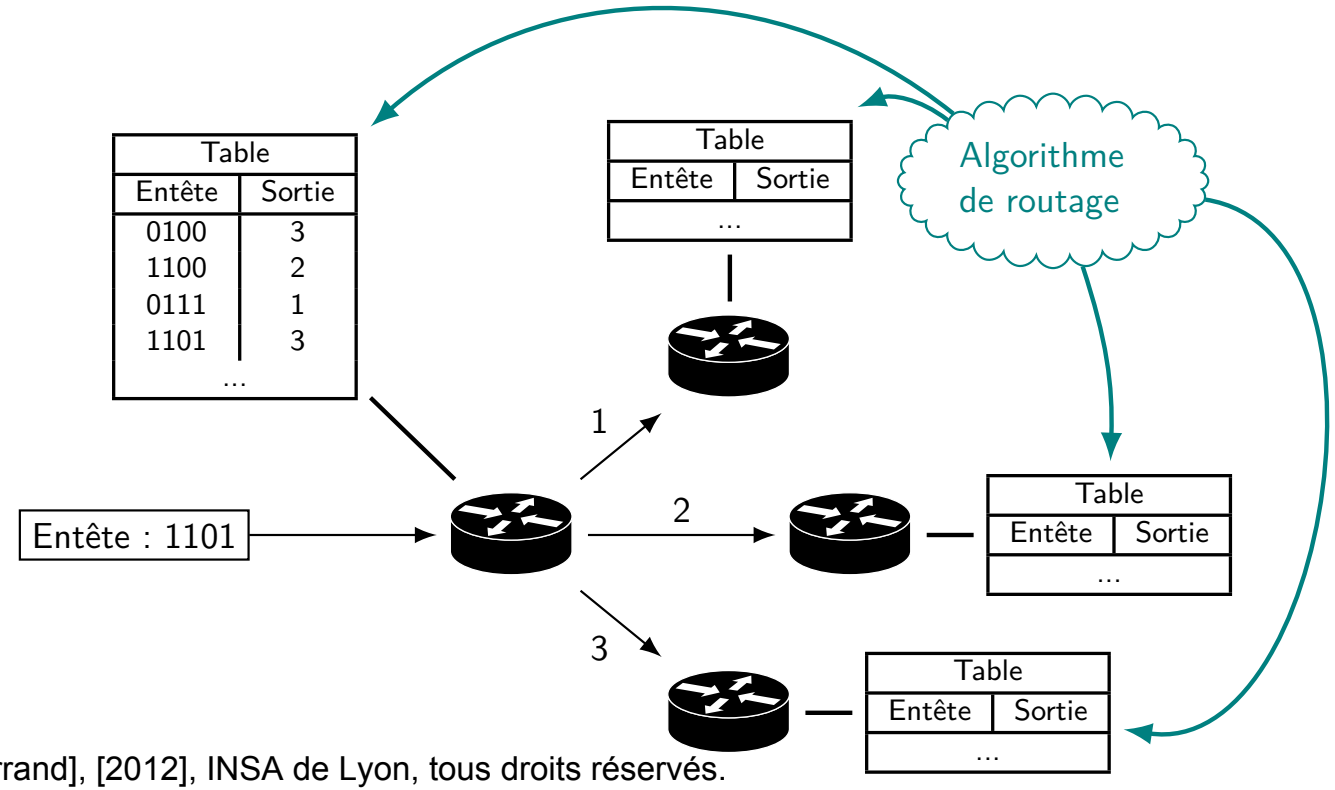


Couche réseau

- La topologie logique de la couche réseau « copie » la topologie entre les **routeurs**
  - Qui sont les éléments centraux de tous les réseaux
- Deux fonctions sont remplies par les routeurs
  - Le **forwarding** (faute de terme français adapté...) des paquets entrants vers un lien de sortie
  - Le **routage**, c'est à dire le choix des routes vers les destinations
- Ces deux termes sont souvent interchangés, mais leur signification est précise
  - La commutation est l'action de base des routeurs (et des switchs que nous verrons plus en avant), où les paquets sont redirigés sur des trajets spécifiques dans le réseau
  - Le routage s'effectue en général en amont ; les routeurs communiquent entre eux pour déterminer quel est le trajet le plus cours entre une source et une destination

Couche réseau

- Chaque routeur possède une **table de *forwarding***, qu'on appelle en français *table de routage*
  - En fonction des informations contenues dans l'entête de la couche réseau, le routeur choisit le point de sortie à l'aide de cette table
  - Elle est remplie à la main, ou plus généralement par un **algorithme de routage**



Services offerts par la couche réseau

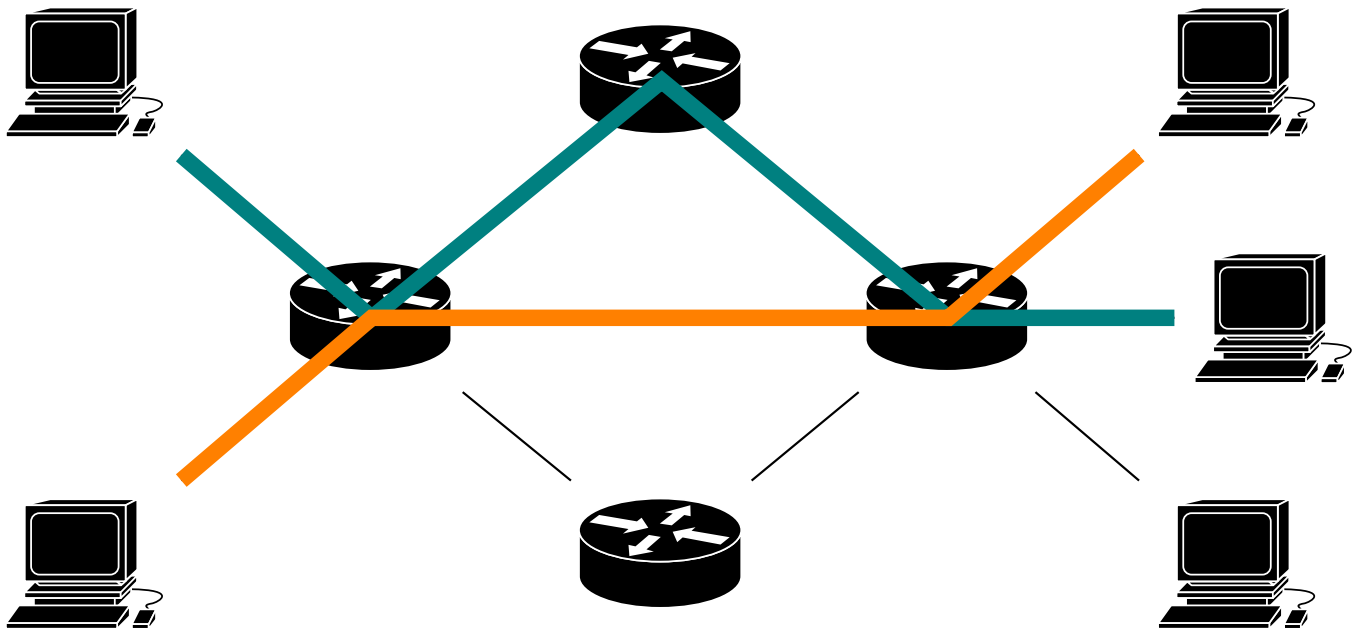
- Sur chaque paquet
  - Arrivée garantie (à un moment donné)
  - Arrivée garantie *en temps limité* (notion de temps-réel)
- Sur un flux de paquets
  - Arrivée des paquets *dans l'ordre*
  - Débit minimal offert, ce qui au passage offre aussi un délai constant
  - Ecart minimal entre deux paquets – en anglais *jitter*
  - Éventuellement de la sécurité

Protocole	Débit garanti	Arrivée garantie	Ordre d'arrivée	Jitter
IP (Internet)	Non	Non	Variable	Variable
ATM (CBR)	Oui, constant	Oui	Dans l'ordre	Constant
ATM (ABR)	Oui, minimum	Non	Dans l'ordre	Variable

- Le protocole IP fournit un **service minimum** !

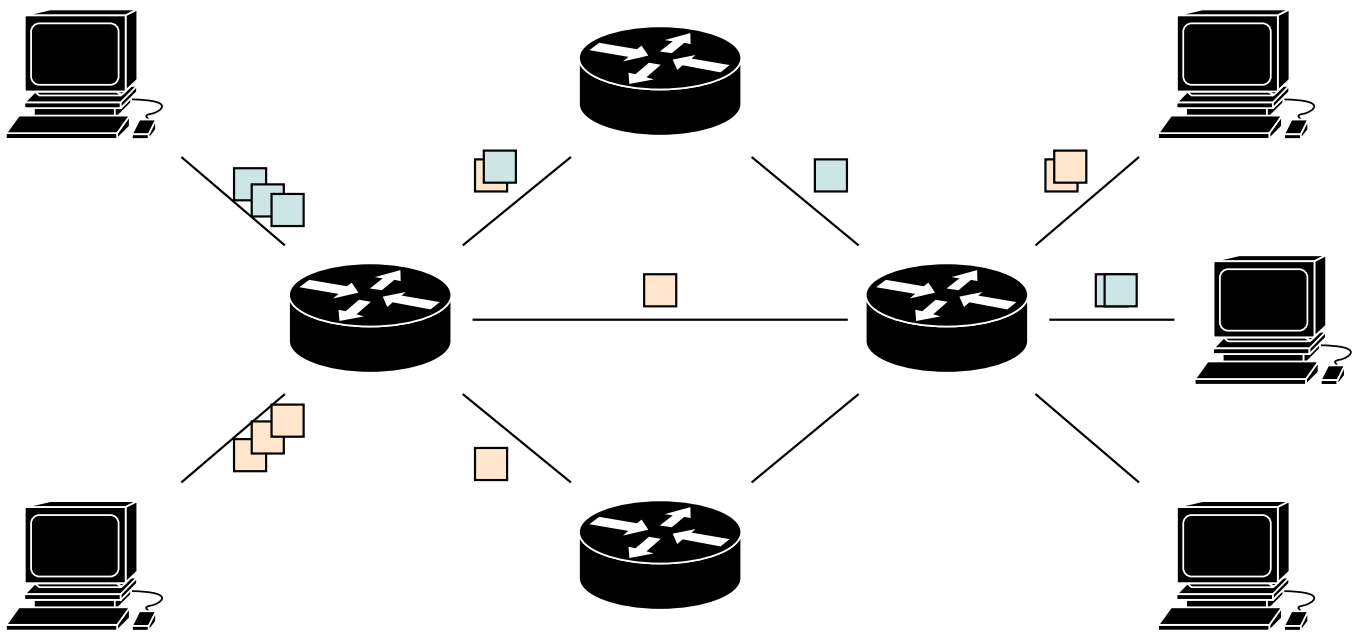
Fonctionnement en mode circuit

- Héritage des réseaux de téléphonie
  - Forme moderne sous le protocole **ATM**, utilisé par les opérateurs pour les réseaux dits *de transmission*, et les lignes spécifiques louées aux entreprises
- Chaque connexion de bout en bout emprunte une route *fixe* et réservée
  - Partage du réseau entre les utilisateurs fait de manière organisée



Fonctionnement en mode paquet

- Le fonctionnement en mode circuit implique un réseau *complexe*, loin de l'idée d'Internet où le réseau doit être le plus stupide possible
- Le protocole **IP** fournit un service minimal, et permet de produire des routeurs à faible coût
- Les paquets transitent sur le réseau de manière désordonnée



Forwarding et modes de fonctionnement

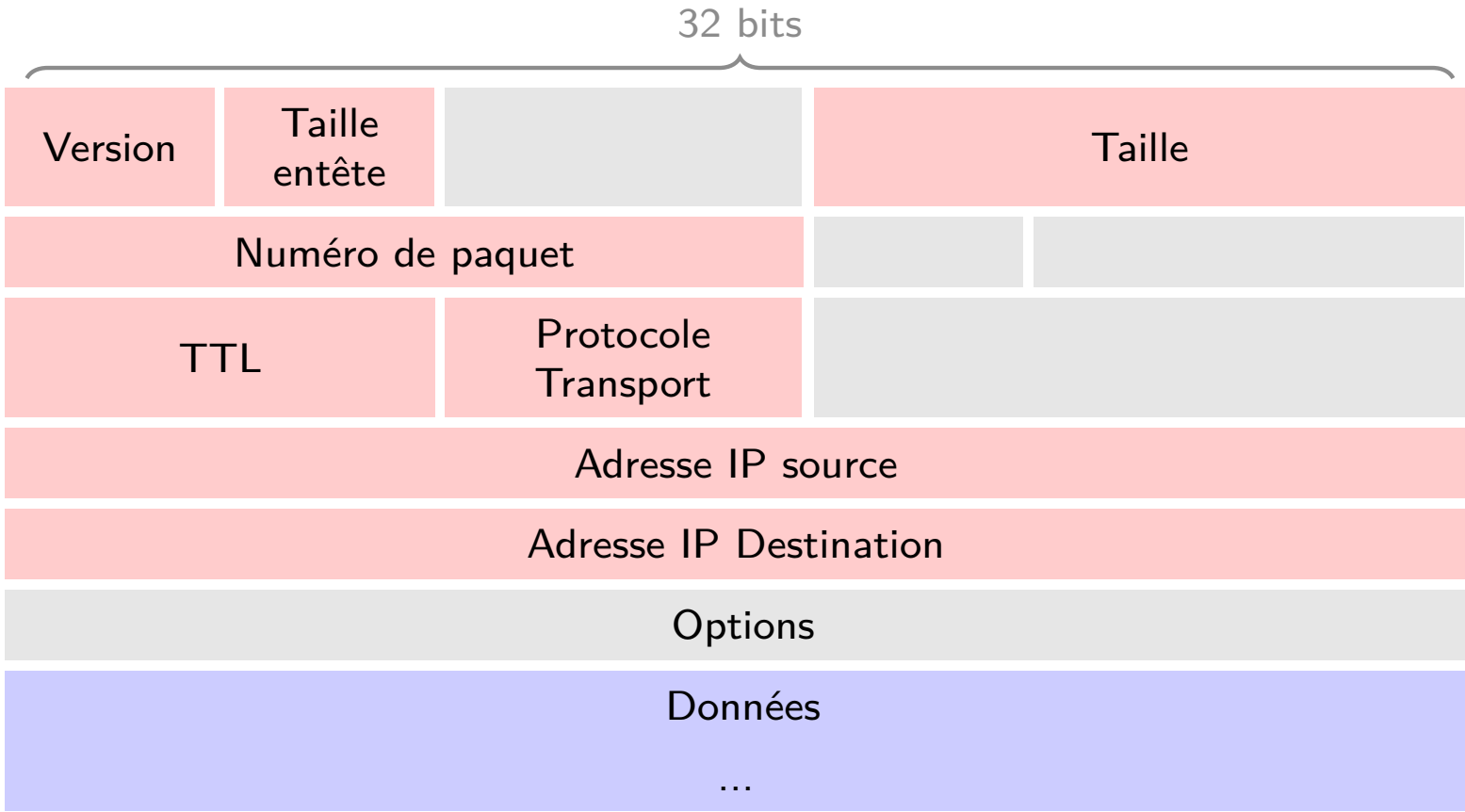
- En mode circuit, la table de routage est mise à jour chaque fois qu'une nouvelle connexion est demandée, et l'entrée correspondante est détruite à la déconnexion
  - L'information emprunte donc toujours le même trajet dans le réseau tant que la connexion est active
  - On parle donc d'un mode **connecté**, comme pour TCP
- En mode paquet, l'algorithme de routage va mettre à jour la table de routage de manière lente
  - Les paquets n'emprunteront donc pas forcément la même route
  - Et ils n'arriveront pas non plus dans l'ordre !
  - On parle d'un mode **non-connecté**
- Le mode paquet a énormément d'intérêts, entre autres :
  - Il gère mieux différents médium physiques de transmission
  - Il permet un partage simple et naturel du réseau entre plusieurs utilisateurs

Le protocole IP

- Le protocole IP a trois grandes composantes
  - Des **sous-protocoles de routages**, que nous ne traiterons pas dans ce cours, mais qui permettent de maintenir à jour les tables de routage dans des réseaux locaux et sur Internet
  - Des **conventions d'adressage** des hôtes sur Internet et dans les réseaux locaux, ainsi que les règles liées au *forwarding* des paquets
  - Un **protocole de contrôle** et de diagnostic appelé **ICMP**, que vous avez peut être déjà utilisé sans le savoir !
- Le protocole IP possède plusieurs versions
  - La version courante est la version 4 (**IPv4**)
  - La version qu'on essaye désespérément de faire passer est la version 6
  - **IPv6** apporte énormément d'améliorations, mais les équipements doivent être mis à jours
  - Entre autres, IPv6 augmente l'espace d'adressage d'IPv4 qui est quasi saturé à ce jour

Datagrammes IP

- Les paquets IP sont souvent appelés des **datagrammes**
- La taille de l'entête IP est au minimum de 20 octets, parfois plus
  - A rajouter à l'entête TCP qui est également de 20 octets
  - Notion de *surdébit* (traduction officielle du terme *overhead*...)



Adressage IP

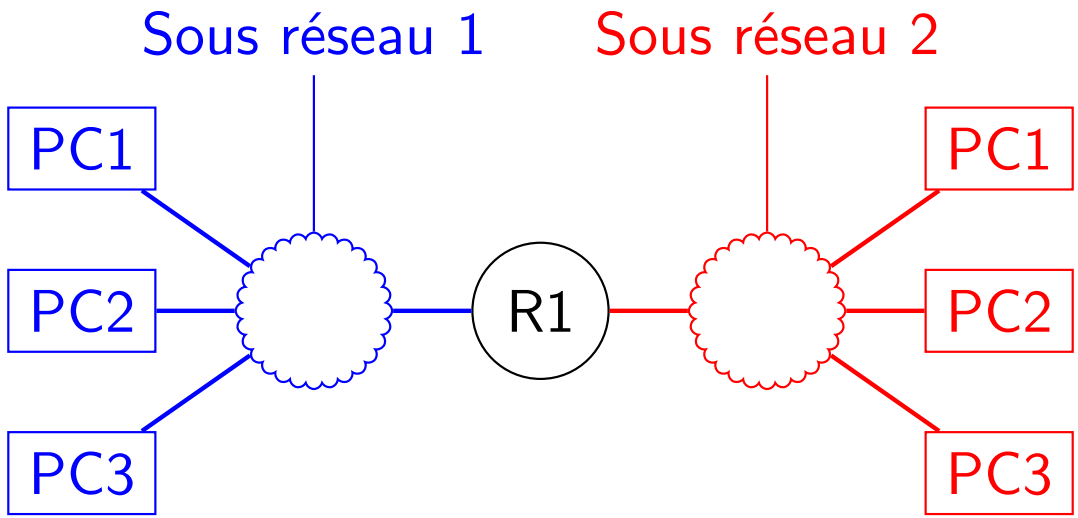
- Une adresse IP est un identifiant **unique** sur Internet
- Chaque **interface de connexion** possède *au moins* une adresse IP
- Un **routeur** a nécessairement plusieurs interfaces (sinon il est inutile) et donc plusieurs adresses
- Une adresse IP est composée de 4 octets, soit 32 bits :

$$\texttt{223.1.1.1} = \underbrace{\texttt{11011111}}_{223}.\underbrace{\texttt{00000001}}_{1}.\underbrace{\texttt{00000001}}_{1}.\underbrace{\texttt{00000001}}_{1}$$

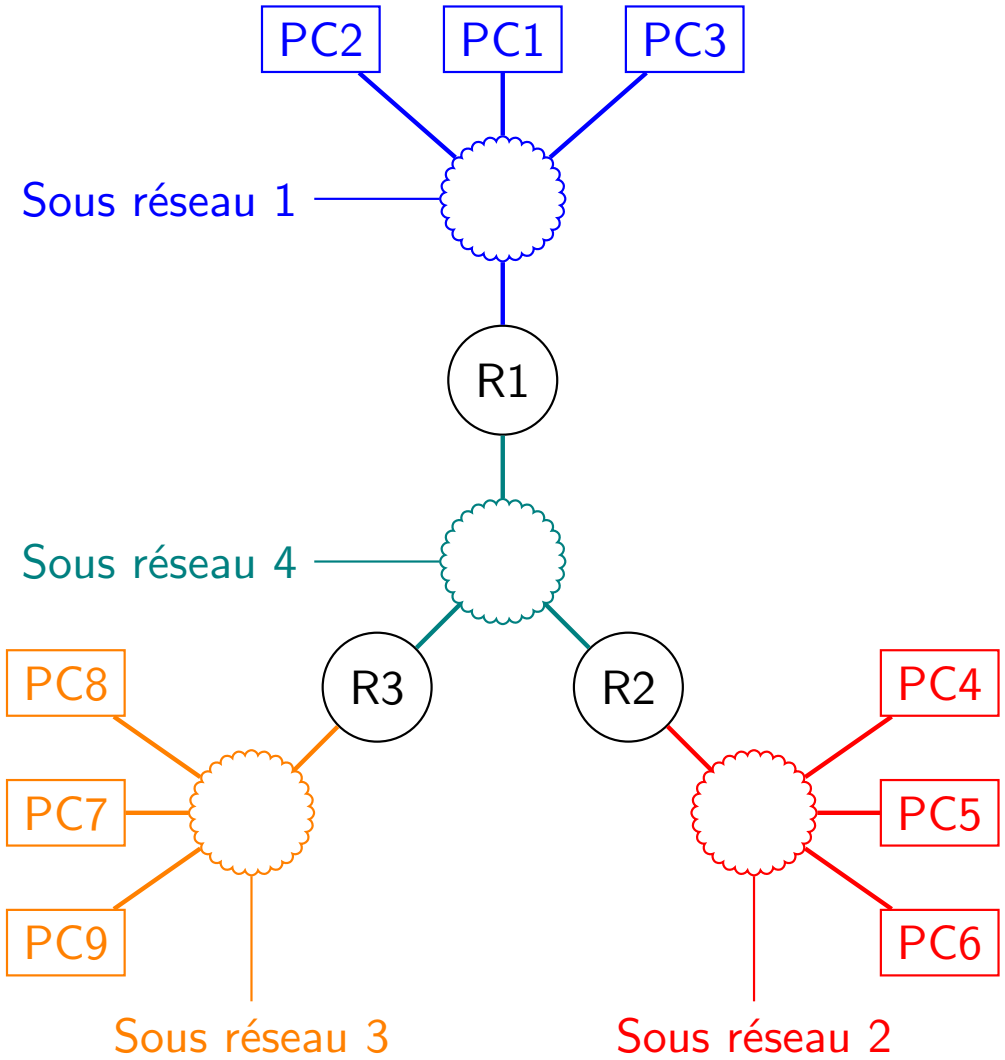
- Les adresses IP sont distribuées par une organisation mondiale qui s'appelle l'**ICANN**
  - Mais elle ne distribue pas les adresses individuellement, mais plutôt par **préfixe de sous-réseau**
- Chaque adresse est séparée en deux composantes *au niveau binaire* :
  - Les bits de poids fort adressent le **sous-réseau**
  - Les bits de poids faible adressent l'**hôte** à l'intérieur du sous-réseau

Adressage IP

- Un sous-réseau est en pratique un ensemble d'hôtes et **une interface** d'un routeur
- Un routeur appartient donc nécessairement à plusieurs sous-réseaux
- Nous verrons plus tard les équipements reliant les hôtes dans un sous-réseau



Adressage IP



Adressage IP

- Pour identifier les sous-réseaux, il suffit d'enlever tous les routeurs !
- Les adresses IP sont attribuées par **préfixes** à des sous-réseaux
  - Le préfixe identifie ainsi le sous-réseau de l'adresse
  - Les fournisseurs d'accès à Internet, ou les structures comme l'INSA, disposent de préfixes dont ils sont libres de faire *ce qu'ils veulent*
  - La gestion des adresses IP individuelles est donc déléguée par l'ICANN

- La notation d'une adresse IP avec son préfixe est la suivante :

$$\underbrace{223.1.1.1/23}_{\text{Préfixe}} = \underbrace{11011111.00000001.0000000}_{\text{23 bits pour le sous-réseau}}\underbrace{1.00000001}_{\text{9 bits pour l'hôte}}$$

- On utilise aussi une notation par **masque de sous-réseau**, en mettant les bits du préfixe à 1 et ceux de l'hôte à 0 :

$$255.255.254.0 = \underbrace{11111111.11111111.1111111}_{\text{23 bits pour le préfixe}}\underbrace{0.00000000}_{\text{9 bits pour l'hôte}}$$

Adressage IP

- Chaque sous-réseau possède donc un préfixe particulier, la seconde partie de l'adresse IP servant à identifier l'hôte dans le sous-réseau
- La première adresse de chaque préfixe est réservée pour devenir **l'adresse de sous-réseau**, qui sert à l'identifier
  - La dernière adresse est également réservée et devient **l'adresse de broadcast**, pour joindre *d'un coup* tous les ordinateurs d'un sous-réseau
- Le masque de sous-réseau permet d'obtenir simplement [1] l'adresse de sous-réseau, par une opération de ET binaire :

```
    11000001.00100000.11011000.00001001 = 134.214.146.5
x   11111111.11111111.11111111.00000000 = 255.255.255.0
-----------------------------------------
    11000001.00100000.11011000.00000000 = 134.214.146.0
```

1. Pour un ordinateur.

Exemples

1. On considère un sous-réseau ayant comme préfixe 128.119.40.128/26. Donnez un exemple d'adresse IP de ce sous-réseau.

2. Combien d'hôtes peut contenir un sous-réseau ayant 28 bits de masque ? Et un sous-réseau ayant 16 bits de masque ?

3. Considérons un réseau utilisant un masque égal à 255.255.248.0.
   a. Donnez le préfixe correspondant à ce masque de sous-réseau
   b. Les 3 stations d'adresses respectives 194.148.208.26, 194.148.216.145 et 194.148.210.32 appartiennent-elles au même sous-réseau ?
   c. Quelle(s) plage(s) d'adresses sont utilisée(s) ? Quelle sont les adresse(s) de diffusion (*broadcast*) de ces plages ?

Plan d'adressage IP

- Les structures disposant d'un préfixe vont devoir les partager entre leurs sous-réseaux
- Par exemple, l'INSA dispose du préfixe 134.214.0.0/16
  - A priori, on peut adresser $2^{16} = 65536$ machines, -2 adresses pour prendre en compte l'adresse de broadcast et de sous-réseau
  - Mais pour des raisons pratiques, on veut diviser ce préfixe en plusieurs sous-réseaux
- Le partitionnement d'un préfixe est une étape centrale dans la construction d'une architecture réseau, et s'appelle un **plan d'adressage**
- Une division simple consiste à choisir des préfixes multiples de 8, ce qui revient à « couper » les adresses IP au niveau des points
  - Dans mon exemple précédent, si je choisis un préfixe en /24, j'aurais les sous-réseaux suivant :
    - 134.214.0.0/24 (Ex d'IP : 134.214.0.4/24)
    - 134.214.1.0/24 (Ex d'IP : 134.214.1.248/24)
    - 134.214.2.0/24 (Ex d'IP : 134.214.2.57/24)
    - ...

Plan d'adressage IP

- Mais il est possible d'être plus précis, et de mieux « coller » au nombre de machines dans un sous-réseau
- En ajoutant 1 bit au préfixe, j'augmente le nombre de sous-réseaux de 2, et je divise la taille de chacun par 2
- Exemple :

<table>
<tr><td colspan="4">134.214.0.0/16</td></tr>
<tr><td colspan="2">134.214.0.0/17</td><td colspan="2">134.214.128.0/17</td></tr>
<tr><td>134.214.0.0/18</td><td>134.214.64.0/18</td><td>134.214.128.0/18</td><td>134.214.192.0/18</td></tr>
</table>

- A chaque étape, il est également possible de diviser seulement une partie des sous-réseaux

<table>
<tr><td colspan="3">134.214.0.0/16</td></tr>
<tr><td colspan="2">134.214.0.0/17</td><td>134.214.128.0/17</td></tr>
<tr><td>134.214.0.0/18</td><td>134.214.64.0/18</td><td>134.214.128.0/17</td></tr>
</table>

Plan d'adressage IP

- Il n'est par contre pas possible d'obtenir la division suivante :

<table>
<tr><td colspan="3">134.214.0.0/16</td></tr>
<tr><td colspan="2">134.214.0.0/17</td><td>134.214.128.0/17</td></tr>
<tr><td>134.14.0.0/18</td><td>134.14.64.0/17</td><td>134.214.192.0/18</td></tr>
</table>

- Une fois divisés, les préfixes sont attribués aux sous-réseaux
  - On attribue ensuite des adresses à chaque hôte, et à chaque interface de routeur
- Comme les routeurs, les hôtes possèdent leur propre **table de routage** qu'il faut renseigner
- Les routeurs connaissant les sous-réseaux auxquels ils sont connectés, ils peuvent directement effectuer un **forwarding**
  - Pour les autres sous-réseaux qui sont reliés à d'autres routeurs, il faut **renseigner la table de routage**, en indiquant le sous-réseau visé et l'adresse IP du prochain saut
  - Pour se simplifier la vie, on utilise beaucoup le principe de la **route par défaut**, si l'on n'en trouve pas d'autres

Plan d'adressage IP

- Si l'on n'est pas riche (ou si on est un particulier), on ne dispose en général pas d'un préfixe *public* d'Internet
- Il est néanmoins possible de créer un réseau local en utilisant des **plages d'adresses privées**
- Ces plages sont au nombre de 3 :
  - 192.168.0.0/16
  - 172.16.0.0/12
  - 10.0.0.0/8
- Il est possible de subdiviser ces plages à volonté comme pour tous les préfixes
- **Elles ne seront jamais routées sur Internet**
  - Leur utilisation est cantonnée aux réseaux locaux
  - Beaucoup de réseaux d'entreprises, et tous les réseaux personnels utilisent ces adresses
- Il faut mettre en place d'autres mécanismes au niveau du routeur pour accéder à internet à partir d'un réseau privé

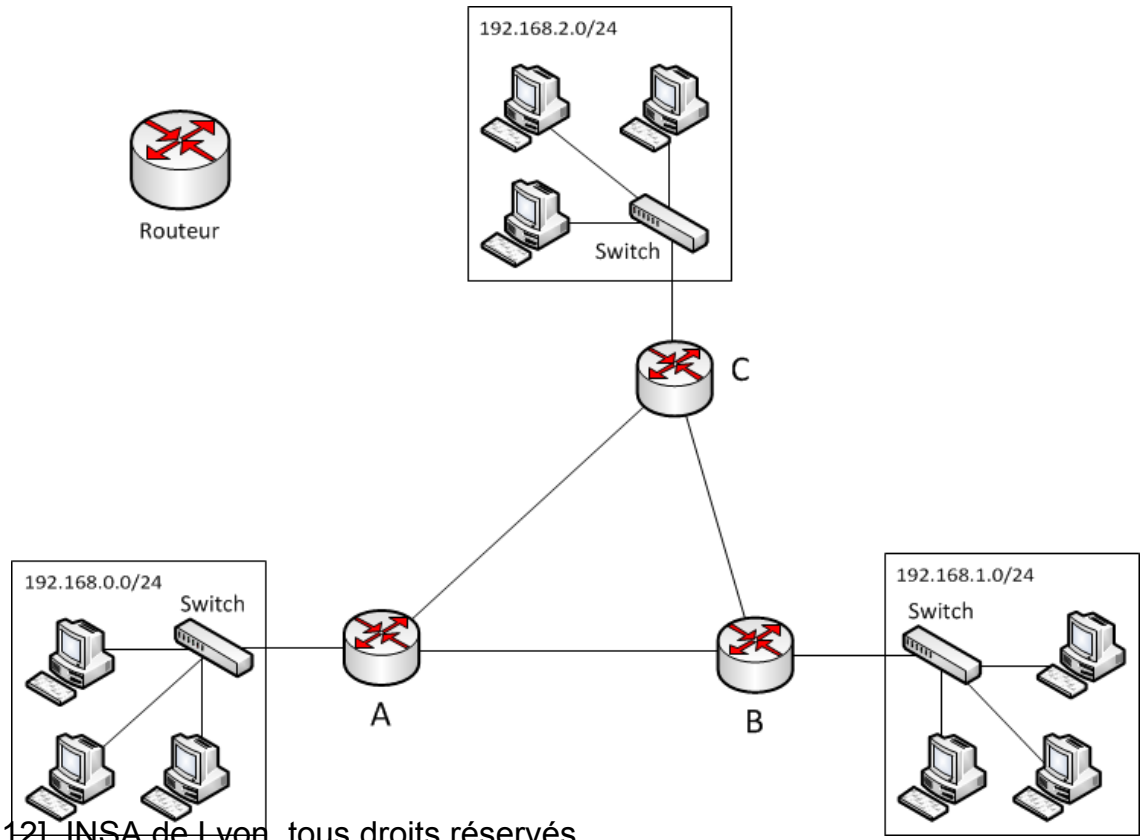
Exercices

1. On considère un FAI possédant le bloc d'adresse indexé par 128.119.40.64/25. Le FAI veut former 4 sous-réseaux de même taille à partir de ce bloc. Indiquez les préfixes (sous la forme a.b.c.d/x) de chaque sous-réseau.
2. Vous disposez du préfixe 192.168.0.0/16, que vous devez subdiviser pour supporter les sous-réseaux suivants :
   - Un sous-réseau A contenant 2050 machines
   - Un sous-réseau B contenant 430 machines
   - 3 sous-réseaux C, D et E, contenant chacun 120 machines

   a. Indiquez les préfixes de chaque sous-réseau, ainsi que le processus que vous mettez en oeuvre pour réaliser le plan d'adressage
   b. Quelle sera la table de routage de votre routeur central ?
   c. *Sans utiliser de route par défaut*, indiquez la table de routage d'un hôte de chaque sous réseaux A, B et E, de telle sorte qu'ils soient capables de communiquer entre eux.

Exercices

1. Considérons le schéma ci-après. Vous disposez du préfixe 10.0.0.0/8 pour relier entre eux les routeurs, que vous pouvez subdiviser comme vous le souhaitez.
   a. Indiquez les adresses IP et le masque de chaque branche pour chaque routeur.
   b. Quelle va être la table de routage du routeur A ? Ressemble-t-elle a celles des routeurs B et C ?
   c. Quelle sera la table de routage d'un ordinateur du sous-réseau en 192.168.0.0/24 ?

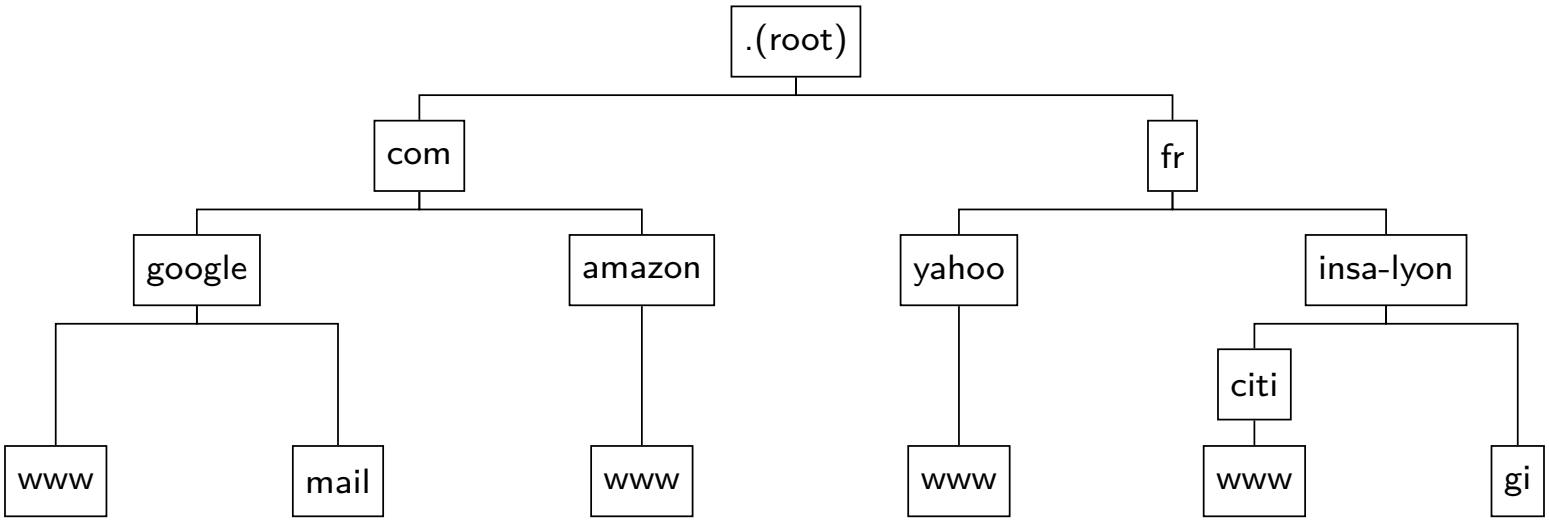


Service de résolution de noms

- *Domain Name Service* : premier « grand » service d'infrastructure
- **Eviter d'utiliser les adresses IP pour adresser les ordinateurs**
  - Fournir aux applications et aux utilisateurs un moyen plus sympathique de repérer une machine sur le réseau local, ou internet
  - *Traduction* d'un **nom d'hôte** en adresse IP :
    - `www.google.fr -> 173.194.34.63`
    - `toonab.citi.insa-lyon.fr -> 134.214.146.38`
- Service implémenté **dans la couche application**
  - Violation du modèle en couche ?
  - Utilise le modèle client-serveur, mais *sans interaction avec l'utilisateur*
- Permet d'identifier les hôtes, mais également les services
  - `toonab.citi.insa-lyon.fr` est un **hôte** du **domaine** insa-lyon.fr
  - `paul.ferrand@insa-lyon.fr` est un utilisateur mail sur le **domaine** insa-lyon.fr
  - Dans ce dernier cas, le DNS fournit le nom d'hôte du *serveur mail*, mais ne contient pas directement l'information sur `paul.ferrand`

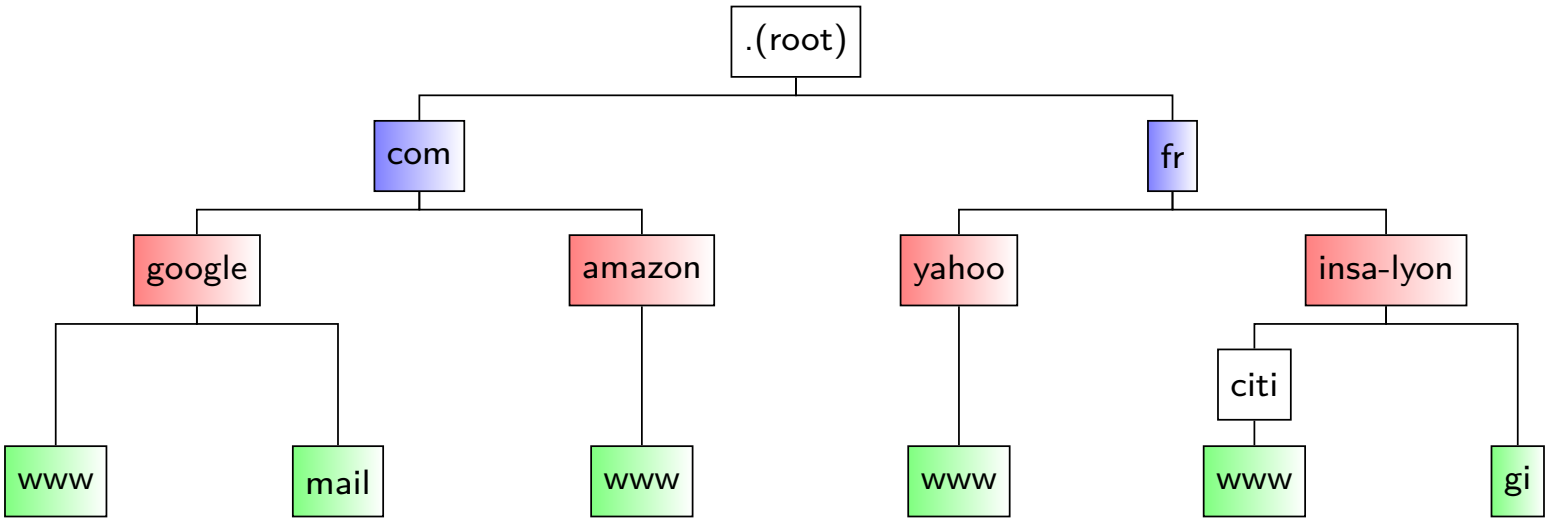
Service de résolution de noms

- Utilise le principe classique de requête-réponse, utilisable en *boîte noire* pour les utilisateurs et les applications
  - Requête : « Qui est www.google.fr »
  - Réponse : « www.google.fr est à l'adresse IP 173.194.34.63 »
  - Géré par le système d'exploitation
- Organisation hiérarchique de la résolution des noms
  - Chaque niveau de l'arbre correspond à un serveur, le tout premier niveau étant le « root »
  - Redondance pour chaque niveau (plusieurs roots, plusieurs serveurs `.com`, ...)



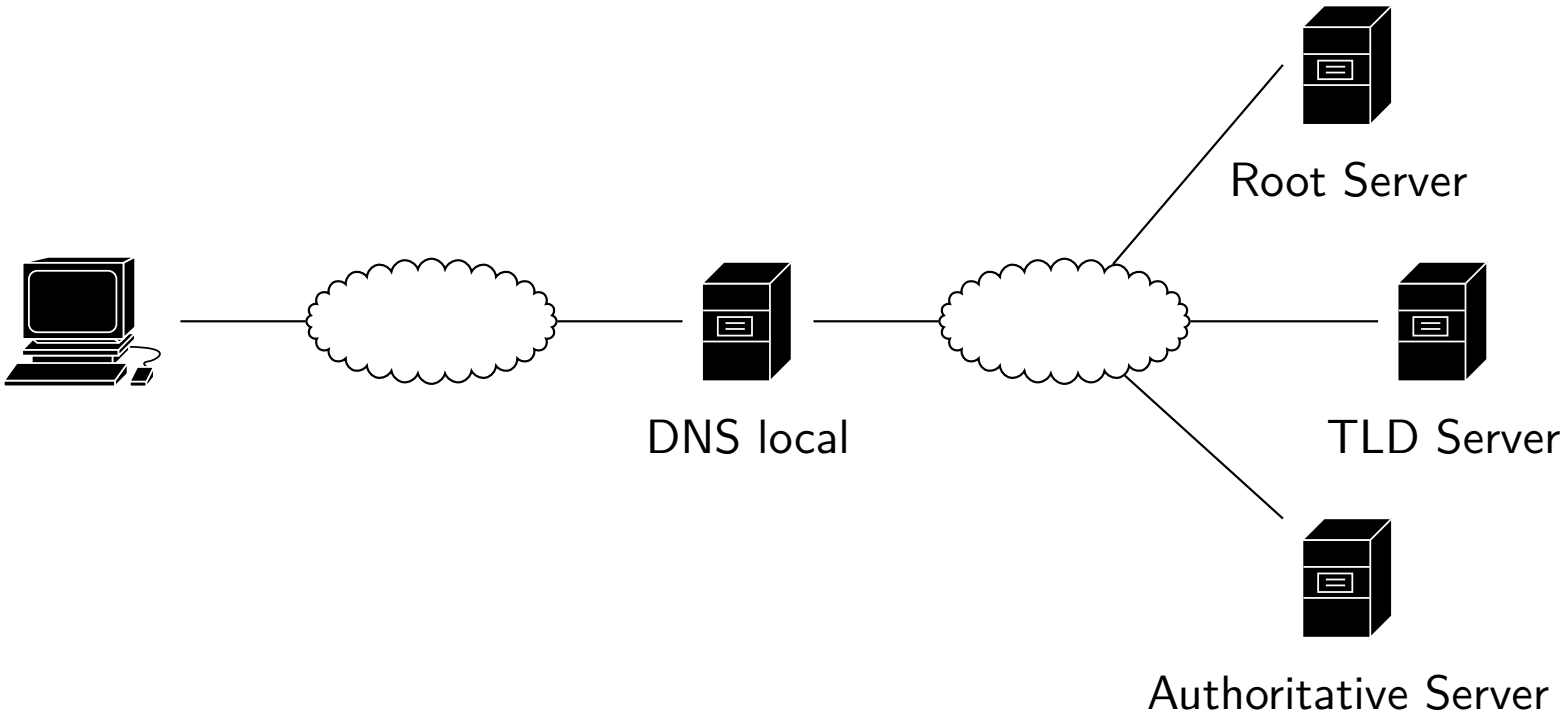
Service de résolution de noms

- A ce jour, environ 250 serveurs *root*
  - Géographiquement répartis, et regroupés en 13 serveurs virtuels accessibles via `a.root-servers.net`, `b.root-servers.net`, `...`
  - Le DNS est un service critique de l'Internet, et ses attaques paralyseraient le monde entier
- Les serveurs liés aux `.com`, `.fr`, `.net`, `.org`, `...` sont appelés **Top-Level Domains Servers** (en bleu)
- Les serveurs liés contenant des adresses d'hôtes sont appelés **Authoritative Servers** (en rouge)



Service de résolution de noms

- Pour un client DNS (un hôte), requête à un **serveur DNS local**, puis
  - Requêtes en direct du serveur local
  - Requêtes récursives depuis le serveur local



Service d'attribution d'adresses IP

- **Dynamic Host Configuration Protocol**
- Protocole *plug-and-play* pour la configuration réseau des ordinateurs
  - Les utilisateurs l'aiment, les administrateurs aussi !
- Utilisation du concept de **diffusion**, en anglais *broadcast*
  - Envoi de paquet à l'ensemble des hôtes d'un *sous-réseau*
  - Echange d'informations communes
- Le DHCP est souvent implémenté dans les routeurs, vous en avez un chez vous sans le savoir
- Fonctionnement en 4 temps, *toujours en diffusion*
  - **DHCP Discover** : requête du client pour faire savoir son arrivée
  - **DHCP Offer** : réponse des serveurs pour offrir une adresse
  - **DHCP Request** : requête du client sur une des adresses offertes
  - **DHCP ACK(nowledge)** : réponse du serveur concerné pour confirmation

Service d'attribution d'adresses IP

- Va plus loin que les adresses IP
  - Information de routage (quelle est ma route par défaut ?)
  - Information sur les DNS (quel serveur DNS utiliser ?)
  - Information sur la sécurité (chiffrement, authentification, ...)
  - Etc
- Fonctionne en utilisant le protocole UDP, sur une couche IP
  - Avant l'attribution, le client a l'adresse 0.0.0.0
  - L'adresse de diffusion **générale** pour le protocole IP est 255.255.255.255
- Les serveurs DHCP peuvent être configurés avec une *mémoire*
  - En utilisant l'adresse physique de la machine, toujours lui donner la même IP
  - Interdire la configuration de machines que le serveur ne connait pas
  - Remplir automatiquement une base DNS avec un nom de machine
- **Élement crucial pour la gestion *centralisée* du réseau**

Couche liaison

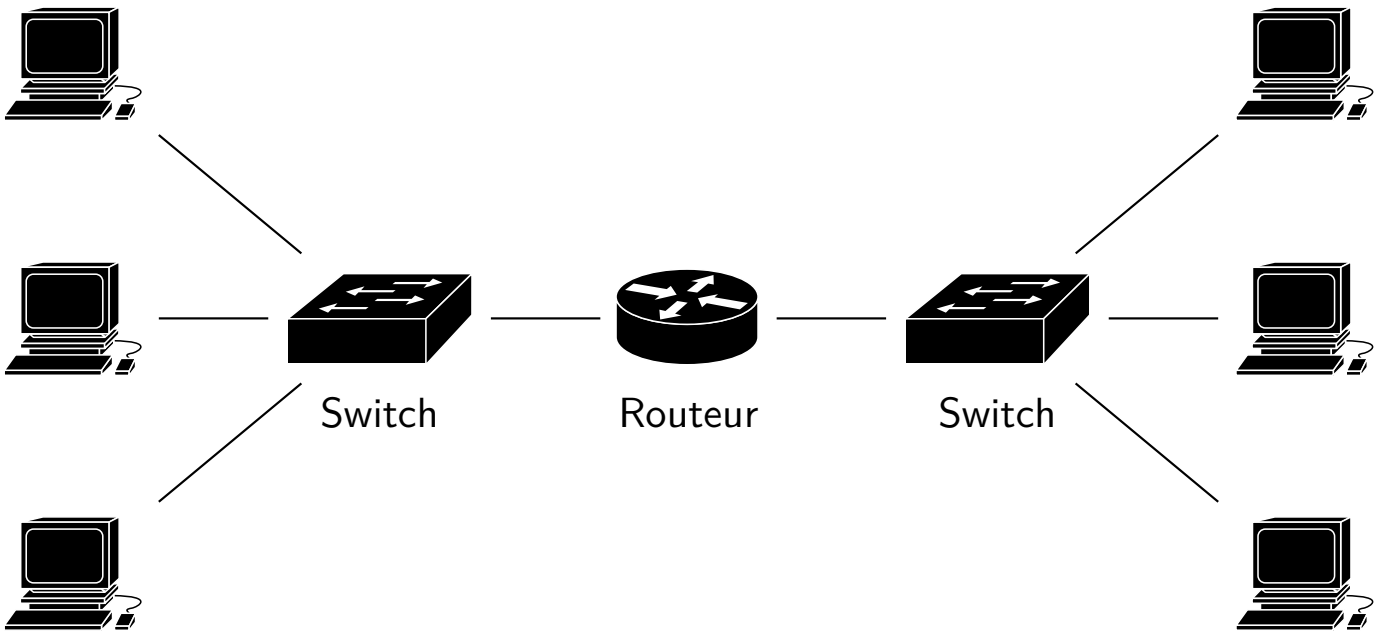
- Dernière couche du modèle présentée dans ce cours
  - La couche physique correspond à la transmission physique des données et aux médiums de transmission
- La couche liaison gère des **noeuds**, qu'ils soient hôtes ou routeurs
  - Représentation en graphe
  - La couche liaison assure les communication sur les arcs, entre les noeuds
- Responsable de
  - La transmission inter-noeud
  - Détection et correction d'erreur
  - Partage du médium de transmission au besoin (WiFi)
- Contrairement à IP et TCP, le protocole de couche liaison est intimement lié à la couche physique
  - Couche *d'abstraction* du médium physique
  - Implémenté directement dans les cartes réseau

Couche liaison

- Assurer la communication **à l'intérieur d'un sous-réseau**, entre les hôtes et le routeur, de manière **transparente**
- L'équipement qui assure cette communication est appelé un **switch**
  - Il assure, comme un routeur, une commutation de paquet
  - Mais il le fait *en dessous* de la couche IP
- Besoin d'un adressage supplémentaire
  - On parle d'adresse *physique*, ou *MAC*
  - Dans les réseaux usuels (WiFi et Ethernet), représentée sur 6 octets, soit 2^{48} adresses possibles
  - Notée en *hexadécimal* : `FE-B3-D1-12-3A-33`
  - **Tous les équipements ont une adresse MAC unique**, et **fixée**
  - Si une machine change de sous-réseau, son adresse MAC reste la même
- L'entête de la couche liaison contient donc, au moins, l'adresse MAC de la source et celle de la destination
  - Soit au minimum $6 + 6 = 12$ octets

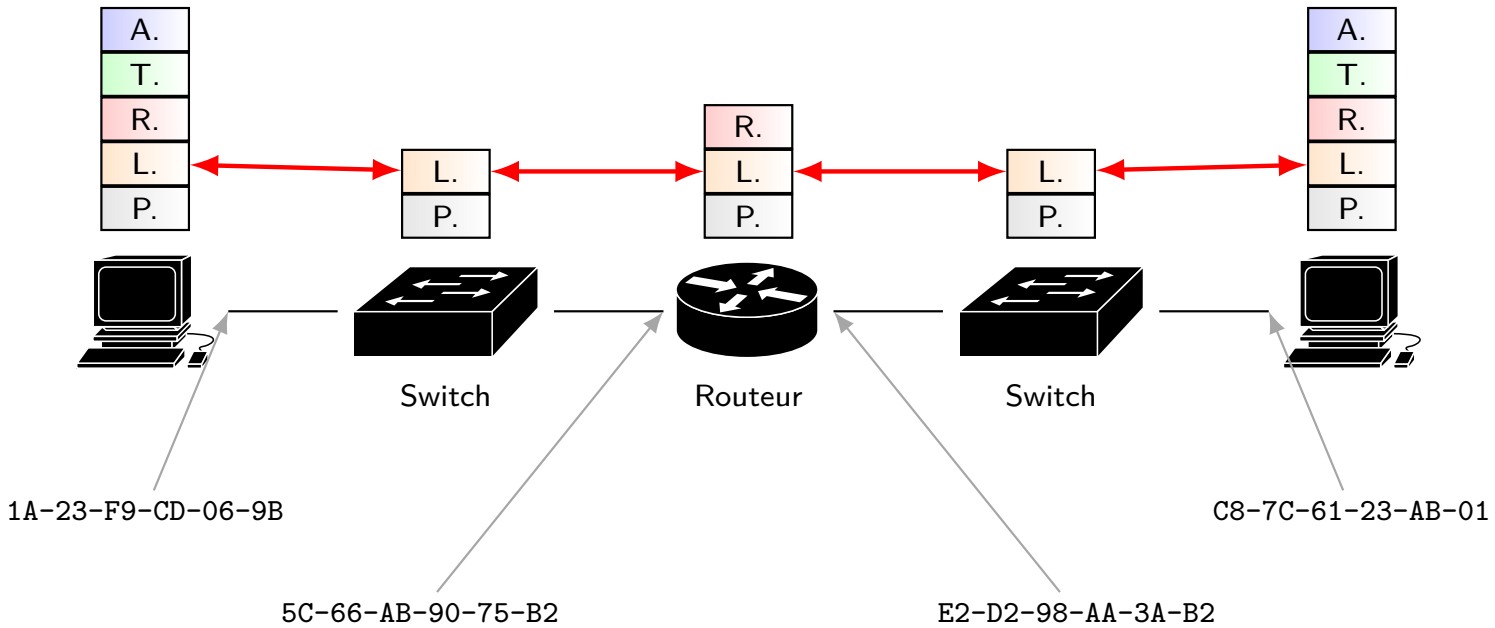
Couche liaison

- Le switch est donc le dernier composant du « nuage » dans les réseaux filaires
  - Tous les réseaux modernes sont constitués d'un assemblage de switchs et de routeurs
- Pour les réseaux sans-fils, la technologie est un peu plus complexe
  - On parle de *ponts* entre les clients sans fils et le réseau général
  - Pas besoin de switch pour que les clients parlent entre eux !



Couche liaison

- Un switch n'a pas d'adresses MAC, mais connait les adresses des équipements qui y sont reliés
- Un équipement possède **une adresse physique par interface**
- A la réception d'un paquet, le switch regarde sa *table de commutation*
  - Si il trouve l'adresse destination, il recopie le paquet sur le bon port
  - Sinon, il demande **en broadcast**, si cette adresse existe
  - Si personne ne répond, il jette le paquet



Couche liaison

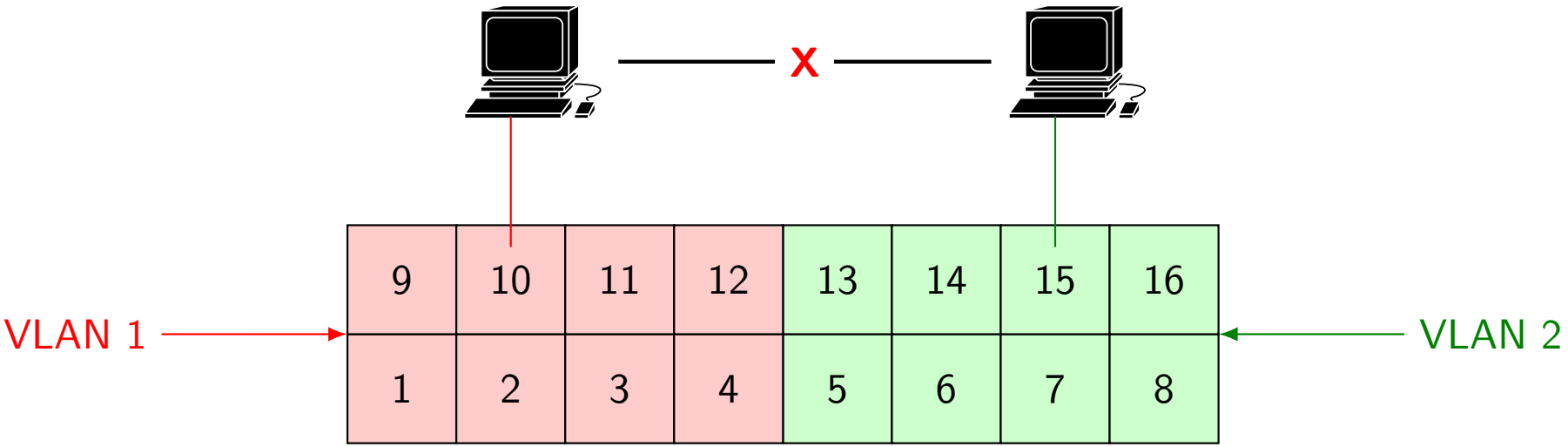
- Pour communiquer sur le sous-réseau, les machines ont donc besoin de connaître les adresses physiques des hôtes connectés
- **ARP (Address Resolution Protocol)**
  - Utilise un méchanisme de requête/réponse
  - La requête est faite par **diffusion** (`FF-FF-FF-FF-FF-FF`), mais pas la réponse
  - Travaille au niveau de la couche 2 (liaison)
  - Pour une IP demandée, renvoie l'adresse MAC
- Les hôtes stockent ces informations dans une table, appelée *table ARP*, et accessible avec la commande `arp -a` sur Windows

Adresse IP	Adresse MAC	TTL
132.214.146.23	`FE-DA-32-32-B3-D6`	`08:23:45`
132.214.146.1	`E2-D2-98-AA-3A-B2`	`08:25:01`
132.214.146.100	`C8-7C-61-23-AB-01`	`08:24:58`

Sous-réseaux virtuels

- Par défaut, les switchs reliés entre eux forment **un, et un seul, sous-réseau**
- La configuration générale « un groupe de switchs par sous-réseau » ne passe pas bien à l'échelle
  - Mauvaise utilisation des ressources
  - Difficulté de gestion et de maintenance
- Pour simplifier la conception d'architecture, et leur reconfiguration, on utilise des **sous-réseaux virtuels** (VLANs)
- Découpage des ports des switchs
  - Chaque port est associé à un identifiant représentant le VLAN auquel il appartient
  - Les hôtes connectés à un VLAN ne peuvent communiquer qu'avec ce VLAN, *en particulier pour la diffusion*
  - Les switchs communiquent entre eux et sont capable de transférer le traffic spécifique à un VLAN vers un autre switch

Sous-réseaux virtuels



- Les switchs peuvent assigner les VLANs par **port**, ou par **adresse MAC**
  - Facilité de configuration, et de maintenance
  - Découpage simple du réseau, indépendamment du nombre de switchs et d'hôtes
- Un hôte sur un VLAN ne peut communiquer qu'avec les hôtes de ce mêmes VLAN
  - Contrôle de la diffusion, en particulier pour ARP qui génère beaucoup de trafic
  - Sécurité par rapport à l'écoute du réseau (voir TP sur l'*ARP Poisoning*)

Architecture logique de réseau

- Première étape de la conception d'un réseau
  - Découpage « logique » du réseau
  - Identification de topologies générale et des équipements à mettre en oeuvre
  - Une architecture claire est plus sécurisée, et permet une meilleure maintenance
- En sortie, un schéma d'architecture logique, comprenant les groupes de machines dans leurs **zones réseau**
- Demande **avant tout** du bon sens, et quelques règles
  1. Le réseau reflète une organisation **et** des contraintes technologiques
  2. On regroupe dans un VLAN des hôtes ayant des utilisations similaires
  3. Les serveurs ne sont jamais placés avec les utilisateurs
  4. Pour les serveurs, les données sont séparées de la présentation
  5. Les zones à risque sont identifiées et isolées
  6. Un serveur de données n'a **jamais** un accès frontal à internet

Architecture logique de réseau

- L'architecture logique permet d'identifier
  - Les VLANs, qui sont les zones de réseau, avec plus ou moins de granularité suivant la précision du découpage en zone
  - La topologie des coeurs de réseaux
- L'architecture logique permet de préciser la réalisation technique du réseau
  1. La liste des **équipements de coeur de réseau** (routeurs, switchs, topologie)
  2. La **liste des VLANs** et de leurs hôtes si besoin
  3. Le plan de nommage des hôtes (et l'inscription en DNS)
  4. Le **plan d'adressage**, où les adresses réseaux disponibles sont découpées en sous-réseaux et attribuées aux différents VLAN
  5. La configuration des services réseaux (DNS et DHCP principalement), et l'adressage des *équipements spéciaux* (serveurs, routeurs, imprimantes, ...)
  6. Les flux d'information et leur trajet, afin de les gérer

Élements de sécurité : les firewalls

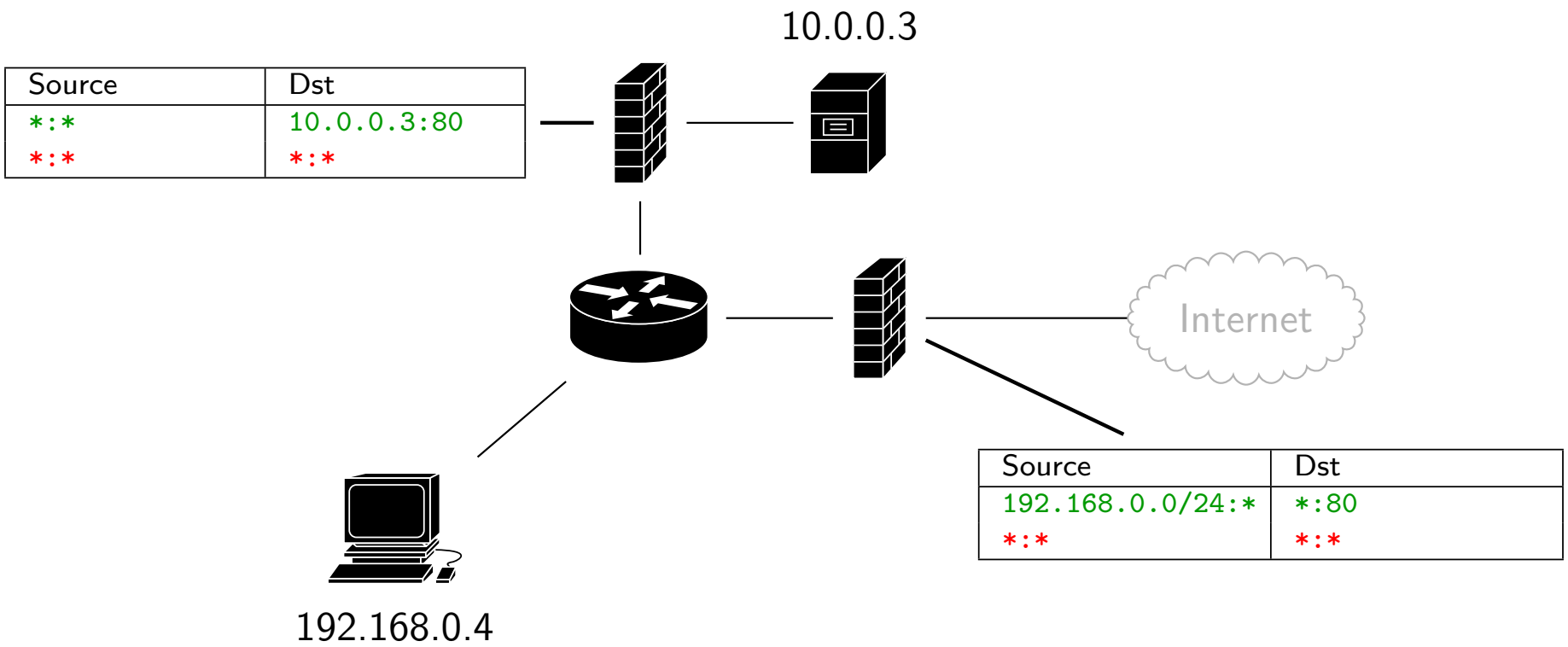
- Un *firewall* est un équipement dont le but est de contrôler les *flux* d'informations dans le réseau
- Toute la subtilité vient donc de la définition des flux
  - Flux entre VLANs, définis par des préfixes IP
  - Flux entre machines, définis par des adresses IP précises
  - Flux entre applications, définis par des adresses IP **et des ports**
  - Flux spécifiques à l'intérieur des applications (filtrage de site web)
- Chaque définition se positionne dans une couche précise de la pile réseau
  - Contrôle par adresses IP et préfixes : *routing filter* (contrôle du routage)
  - Contrôle par adresses IP et par port : *packet filter* (contrôle des paquets)
  - Contrôle fin du protocole TCP : *stateful filter* (contrôle des *connexions*)
  - Contrôle des protocoles applicatifs : *protocol filter* (ou *deep packet inspection*)

Les firewalls

- Le contrôle du routage peut se faire au niveau du routeur
  - Interdire par exemple le routage entre différents VLANs
- Pour les autres contrôles, parfois intégrés au routeur, sinon il s'agit d'équipements supplémentaires
  - De plus en plus chers en fonction de la finesse du contrôle
- *La première étape est donc d'identifier les flux*, par exemple :
  - Tous les utilisateurs → Internet
  - Internet → Un serveur web
  - Les utilisateurs → le serveur de base de données
  - Les utilisateurs → le serveur DNS
  - ...
- Puis de les traduire en termes de *préfixes* et de *ports* :
  - `192.168.0.0/24:*` → `*:80`
  - `*:*` → `10.0.0.4:80`
  - `192.168.0.0/24:*` → `10.0.0.5:1023`
  - `192.168.0.0/24:*` → `10.0.0.2:53`
  - ...

Les firewalls

- Chaque couple `(IP:port)` $\rightarrow$ `(IP:port)` identifie un flux
- Le firewall fonctionne par liste blanche
  - Tous les flux sont interdits, sauf ceux explicitement écrits dans une **table de filtrage**
  - Précédence hiérarchique des règles
  - La dernière règle refuse donc tous le trafic



Les firewalls

- Ensuite, pour chaque application, on peut envisager l'utilisation d'une *passerelle d'application* appelée **proxy**
  - Ces passerelles peuvent comprendre une étape d'autentification
  - Elles peuvent aussi archiver l'historique des connexions
  - Les proxies sont parfois *transparents*
- Exemple d'utilisation de passerelles applicatives
  - Filtrage de sites HTTP
  - Filtrage des mails entrants et sortant, pour éviter le spamming
  - Serveurs d'accès distants, pour l'accès à des applications critiques
- La sécurisation de l'architecture du réseau passe donc par :
  - La définition de **tous les flux autorisés dans le réseau**
  - Le positionnement des firewalls, intégrés au routeur pour les cas simples ou séparés pour des charges lourdes
  - Le choix des passerelles applicatives si besoin, et la *redirection des flux vers ces passerelles*

Bibliographie

Kurose, James S. and Ross, Keith W.
Computer Networking : A Top-Down Approach
Pearson, 6e édition, 2012

Tanenbaum, Andrew S. and Wetherall, David J.
Computer Networks
Prentice Hall, 5e édition, 2010

- Network sprites courtesy of Cisco© : retrieved from `http://www.cisco.com/web/about/ac50/ac47/2.html`
- Images :
  - Arpanet maps : ©Heart, F., McKenzie, A., McQuillian, J., and Walden, D., *ARPANET Completion Report*, Bolt, Beranek and Newman, Burlington, MA, January 4, 1978. Retrieved from `http://som.csudh.edu/cis/lpress/history/arpamaps/`
  - Level 3 network map : ©Level 3, retrieved from `http://www.level3.com`
  - Verizon network map : ©Verizon, retrieved from `http://www.verizonenterprise.com/about/network/maps/map.xml`