

Réseaux

Introduction

E. Jeandel

Emmanuel.Jeandel at univ-lorraine.fr

Réseaux = Réseaux1 + Réseaux2

- Apprendre comment marche Internet
 - Pas forcément les autres réseaux
- Du cambouis (L2) à la programmation (L3)
- L2 :
 - Intro à la théorie du codage
 - Routage IP
 - Pas de programmation (L3), ni de théorie du signal (R&T)

Avant :

- Codage numérique (L1)

Pendant

- Outils Systèmes + Systèmes

Après

- Réseaux 2 (L3) + Sécurité (L3)

- Videos
- Slides vus en cours disponibles sur arche
- Notes de cours dispos sur arche (peut-être)
- Internet et Bibliothèque
 - James F. Kurose et Keith W. Ross Analyse Structurée des Réseaux
 - Andrew Tanenbaum, Réseaux
 - Douglas E. Comer, Réseaux et Internet
 - Richard Stevens, TCP/IP illustré

Notes de cours autorisées à l'examen.

Évaluation

- Une seule note : compte pour 35% de l'UE Systèmes-Réseaux
 - Obtenu avec un examen final, mais sans doute aussi des TDs/TPs notés
 - TDs en présentiel pour celles et ceux qui le veulent
 - TPs en présentiel ou à distance

Contenu

1 Internet

2 Modèle OSI

Internet = Inter + Net

(Interconnexion de Réseaux)

Un réseau contient

- Des liens de communication (le medium)
- Des données qui sont transportées
- Les éléments sources et destination des données
 - Hôtes, Machines (*end systems*)
- Du *matériel réseau* : éléments intermédiaires entre la source et la destination
- Le logiciel réseau

Deux types de media

- Guide physique (electrons, onde lumineuse)
- Sans guide physique (dans l'air)

Deux types de transmission :

- réseau point à point : entre deux machines/équipements
- réseau à diffusion : un seul canal partagé par toutes les machines

Internet = Inter + Net

(Interconnexion de Réseaux)

Plusieurs types de réseaux, suivant la taille :

- Réseaux locaux (LAN : Local Area Network)
- Réseaux métropolitains (MAN : Metropolitan Area Network)
 - Connecte les réseaux locaux entre eux
- Réseaux distants (WAN : Wide Area Network)
 - Connecte les réseaux locaux et métropolitains entre eux
 - En général point à point

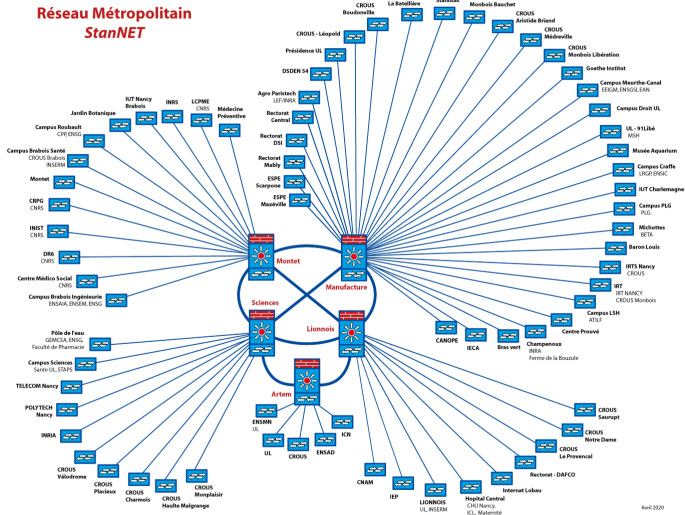
Matériel d'interconnexion : *les routeurs*.

Exemples

Réseaux locaux :

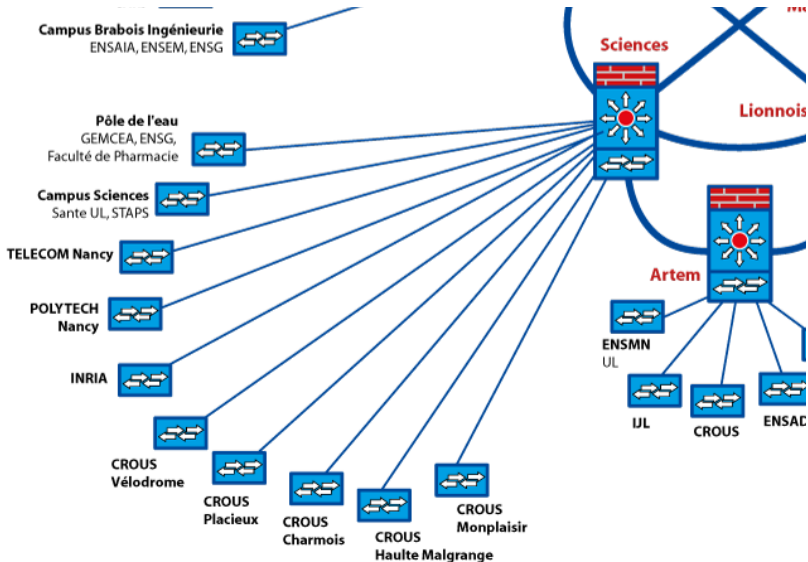
- Réseau chez vous
- Réseau à la FST
- Réseau au Loria

Réseaux métropolitains



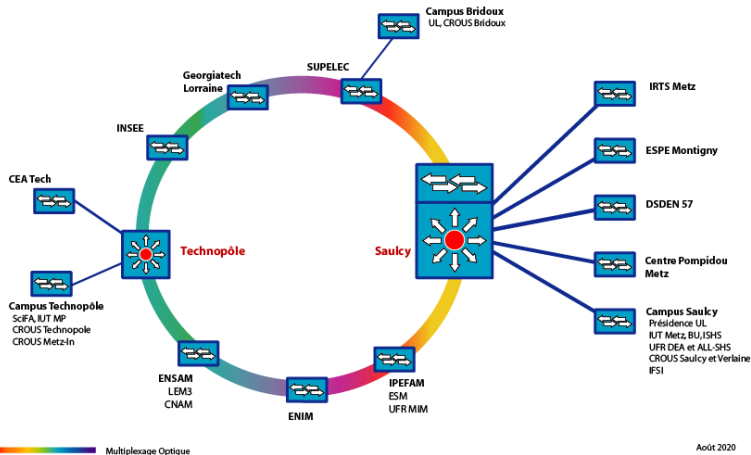
<https://lothaire.univ-lorraine.fr>

Réseaux métropolitains



Réseaux métropolitains

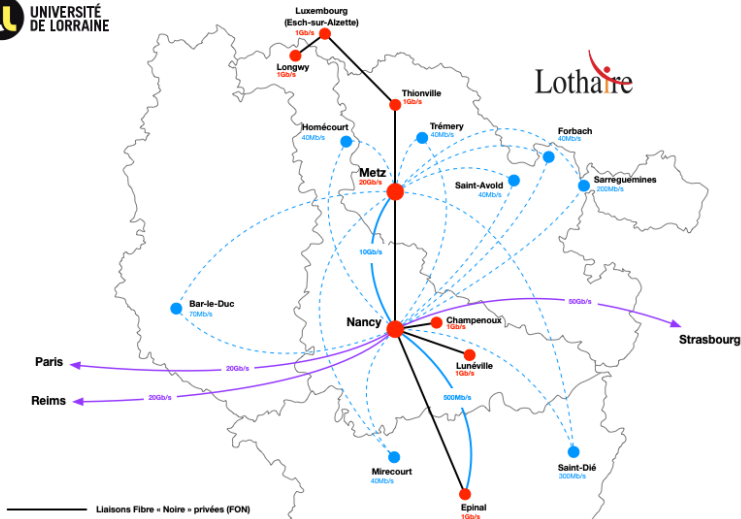
Réseau Métropolitain *AmpereNET*



<https://lothaire.univ-lorraine.fr>

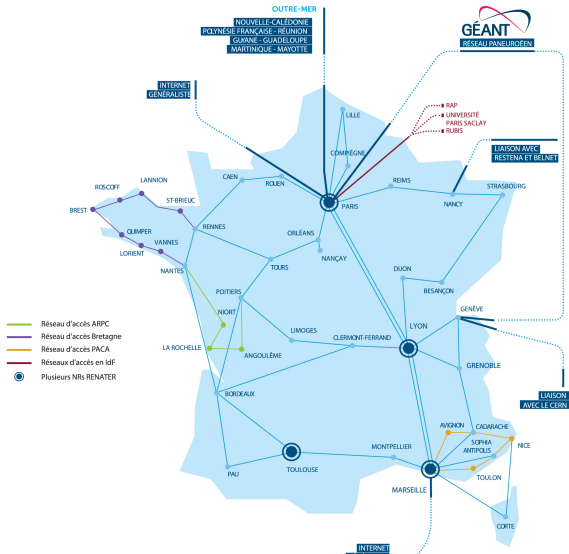
Réseaux WAN

<https://lothaire.univ-lorraine.fr>



Réseaux WAN

<https://www.renater.fr/fr/reseau>



Exemples

Fournisseurs d'accès :

- LAN : réseau chez vous
- MAN : NRA (noeud de raccordement d'abonnés)
- WAN : réseau régional ou national du FAI

Note : Renater, Lothaire sont des FAIs !

Internet = Inter + Net

- Internet = Interconnexion de réseaux locaux *publics* par des MAN et des WAN
- Tous les réseaux locaux qui acceptent d'être connus de tous.
 - Réseau local à la FST fait partie d'Internet
 - Réseau local chez vous ne fait pas partie d'Internet
- Il existe des interconnexion de réseaux qui ne sont PAS publics (ex : armée, grandes entreprises)

- S'assurer de pouvoir faire communiquer des machines sur des réseaux différents.
- Si je me connecte sur l'ENT de la fac depuis chez moi, par où passe le message ?

Tous les fournisseurs d'accès ne sont pas reliés directement à tous les fournisseurs d'accès

- 7000 FAI aux US (source : wikipedia) : il faudrait 25 millions de cable

Hiérarchie dans les fournisseurs d'accès, en 3 niveaux :

- Les FAI transmettent les messages aux FAI de niveaux supérieurs
- Une dizaine de FAI de niveau 1, tous interconnectés

Un exemple

La commande

```
tracert www.univ-lorraine.fr
```

permet d'afficher l'ensemble des routeurs sur le chemin entre mon chez-moi et l'Université de Lorraine

Un exemple

La commande

```
tracert -q1 www.loria.fr
```

permet d'afficher l'ensemble des routeurs sur le chemin entre mon chez-moi et le Loria

Un exemple

```
1 192.168.0.254 (192.168.0.254)
2 194.149.169.89 (194.149.169.89)
3 free-paris-por1.bb.ip-plus.net (193.5.122.58)
4 193.51.187.208 (193.51.187.208)
5 xe-0-0-15-paris1-rtr-131.noc.renater.fr (193.51.177.1)
6 te0-0-0-2-ren-nr-nancy-rtr-091.noc.renater.fr (193.51.177.1)
7 lothaire-vl50-tel-3-nancy-rtr-021.noc.renater.fr (193.51.177.1)
8 gw1-interco-lothaire.loria.fr (193.50.80.14)
```

Un exemple

```
1  192.168.0.254 (192.168.0.254)
2  194.149.169.89 (194.149.169.89)
3  free-paris-por1.bb.ip-plus.net (193.5.122.58)
4  193.51.187.208 (193.51.187.208)
5  xe-0-0-15-paris1-rtr-131.noc.renater.fr (193.51.177.1)
6  te0-0-0-2-ren-nr-nancy-rtr-091.noc.renater.fr (193.51.177.2)
7  lothaire-vl50-tel-3-nancy-rtr-021.noc.renater.fr (193.51.177.3)
8  gw1-interco-lothaire.loria.fr (193.50.80.14)
```

ligne 1 : mon routeur. De 0 à 1, on est sur le réseau local (LAN)

Un exemple

```
1 192.168.0.254 (192.168.0.254)
2 194.149.169.89 (194.149.169.89)
3 free-paris-por1.bb.ip-plus.net (193.5.122.58)
4 193.51.187.208 (193.51.187.208)
5 xe-0-0-15-paris1-rtr-131.noc.renater.fr (193.51.177.131)
6 te0-0-0-2-ren-nr-nancy-rtr-091.noc.renater.fr (193.51.177.91)
7 lothaire-vl50-tel-3-nancy-rtr-021.noc.renater.fr (193.51.177.21)
8 gw1-interco-lothaire.loria.fr (193.50.80.14)
```

ligne 2 : routeur chez mon FAI.

Un exemple

```
1 192.168.0.254 (192.168.0.254)
2 194.149.169.89 (194.149.169.89)
3 free-paris-por1.bb.ip-plus.net (193.5.122.58)
4 193.51.187.208 (193.51.187.208)
5 xe-0-0-15-paris1-rtr-131.noc.renater.fr (193.51.177.131)
6 te0-0-0-2-ren-nr-nancy-rtr-091.noc.renater.fr (193.51.177.91)
7 lothaire-vl50-tel-3-nancy-rtr-021.noc.renater.fr (193.51.177.21)
8 gw1-interco-lothaire.loria.fr (193.50.80.14)
```

ligne 3 : routeur chez Swisscom

Un exemple

```
1 192.168.0.254 (192.168.0.254)
2 194.149.169.89 (194.149.169.89)
3 free-paris-por1.bb.ip-plus.net (193.5.122.58)
4 193.51.187.208 (193.51.187.208)
5 xe-0-0-15-paris1-rtr-131.noc.renater.fr (193.51.177.131)
6 te0-0-0-2-ren-nr-nancy-rtr-091.noc.renater.fr (193.51.177.91)
7 lothaire-vl50-tel-3-nancy-rtr-021.noc.renater.fr (193.51.177.21)
8 gw1-interco-lothaire.loria.fr (193.50.80.14)
```

ligne 4-7 : routeurs chez Renater

Un exemple

```
1 192.168.0.254 (192.168.0.254)
2 194.149.169.89 (194.149.169.89)
3 free-paris-por1.bb.ip-plus.net (193.5.122.58)
4 193.51.187.208 (193.51.187.208)
5 xe-0-0-15-paris1-rtr-131.noc.renater.fr (193.51.177.131)
6 te0-0-0-2-ren-nr-nancy-rtr-091.noc.renater.fr (193.51.177.91)
7 lothaire-vl50-tel-3-nancy-rtr-021.noc.renater.fr (193.51.177.21)
8 gw1-interco-lothaire.loria.fr (193.50.80.14)
```

ligne 8 : routeur chez Lothaire

Problématiques

- Efficacité
- Argent
- Géopolitique

Contenu

- 1 Internet
- 2 **Modèle OSI**

Du départ d'un message à sa réception, il y a beaucoup de tâches différentes à réaliser pour son acheminement.

Modèle par couche

Modèles par couche

Tous les réseaux (abstraits ou concrets) s'appuient sur un **modèle par couche** :

- Chaque couche est en charge d'une tâche bien précise ;
- A l'intérieur d'une machine, chaque couche de niveau n ne peut parler qu'avec la couche $n - 1$;
- Entre deux machines, les couches qui communiquent sont de même niveau. Elles communiquent à travers un **protocole**.

Deux modèles importants :

- Modèle OSI (théorique)
- Modèle TCP/IP (pratique)

Encapsulation

Si la couche numéro 4 d'une machine *A* veut envoyer un message *x* à la couche numéro 4 d'une autre machine *B* :

- Elle passe ce message *x* à la couche 3. La couche 3 va rajouter au message les paramètres dont elle a besoin pour communiquer avec la couche 3 de l'autre machine (qui peut être une autre machine que *B*)
- La couche 3 passe ce message *y* à la couche 2. La couche 2 va rajouter au message les paramètres dont elle a besoin pour communiquer avec la couche 2 de l'autre machine (qui peut être une autre machine que *B*)
- La couche 2 passe ce message *z* à la couche 1. La couche 1 envoie directement le message sur le câble/le wifi/etc.

Du point de vue du récepteur, c'est la même chose dans l'autre sens.

Métaphore des poupées russes

Exemple

Assia veut envoyer un message à Bob. Il demande au module *application* de le faire



Bonjour!

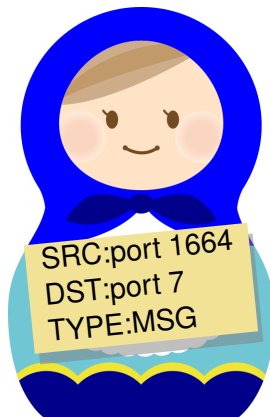
Exemple

Le module application met le message d'Assia dans une poupée, et ajoute les paramètres dont il a besoin pour travailler. Il transmet le message au module *transport*



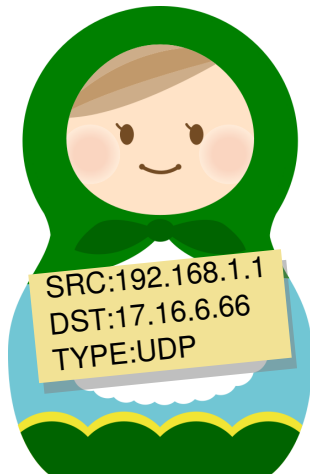
Exemple

Le module *transport* met le message du module application dans une poupée, ajoute les paramètres dont il a besoin pour travailler. Il transmet le message au module *réseau*



Exemple

Le module *réseau* met le message du module transport dans une poupée, ajoute les paramètres dont il a besoin pour travailler. Il transmet le message au module *liaison*



Exemple

Le module *liaison* met le message du module réseau dans une poupée, ajoute les paramètres dont il a besoin pour travailler. Il transmet le message au module *physique*



Exemple

Le module *physique* met tout ça dans une poupée, et envoie sur le cable



Exemple

La poupée arrive sur un *hub* (concentrateur). Un hub ne fait rien d'intelligent, et se contente de cloner la poupée et de la renvoyer tous les autres cables auxquels il est relié



Exemple

La poupée arrive sur un *switch* (commutateur). Le switch enlève la première poupée pour lire les infos du module liaison et savoir sur quel cable envoyer la poupée



Exemple

La poupée arrive sur un *switch* (commutateur). Le switch enlève la première poupée pour lire les infos du module liaison et savoir sur quel cable envoyer la poupée



Exemple

La poupée arrive sur un *switch* (commutateur). Le switch enlève la première poupée pour lire les infos du module liaison et savoir sur quel cable envoyer la poupée



Exemple

La poupée arrive sur un *routeur*. Le routeur enlève les deux premières poupées pour savoir sur quel cable envoyer la poupée



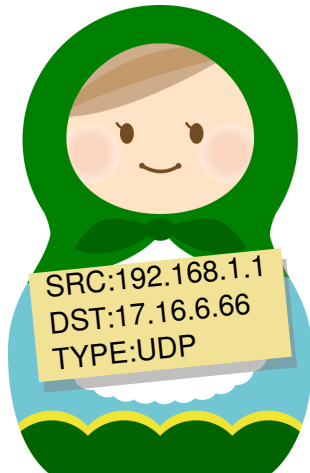
Exemple

La poupée arrive sur un *routeur*. Le routeur enlève les deux premières poupées pour savoir sur quel cable envoyer la poupée



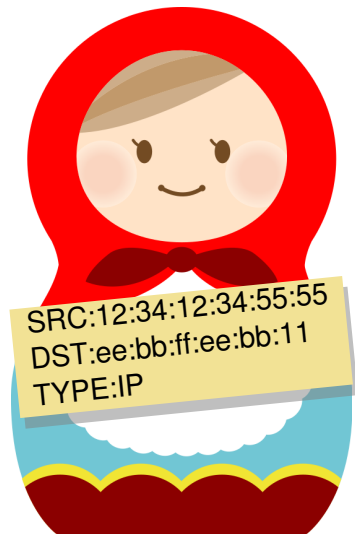
Exemple

La poupée arrive sur un *routeur*. Le routeur enlève les deux premières poupées pour savoir sur quel cable envoyer la poupée



Exemple

La poupée arrive sur un *routeur*. Le routeur enlève les deux premières poupées pour savoir sur quel cable envoyer la poupée



Exemple

La poupée arrive sur un *routeur*. Le routeur enlève les deux premières poupées pour savoir sur quel cable envoyer la poupée



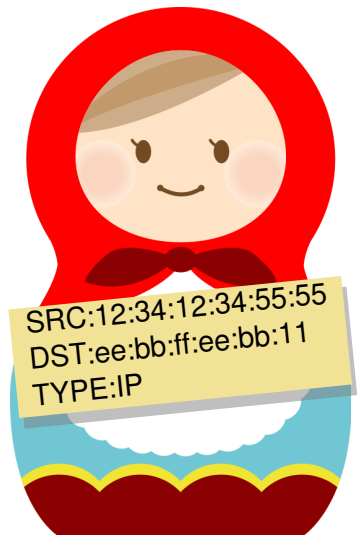
Exemple

La poupée passe par beaucoup d'autres routeurs, hubs et switches, et arrive sur la machine de Bob, qui enlève la première poupée et passe au module liaison



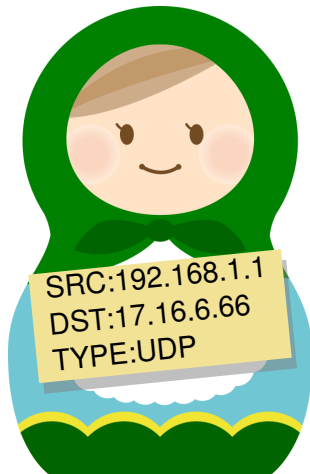
Exemple

Le module liaison lit les paramètres, et en déduit qu'il doit passer la poupée à la fonction IP du module réseau



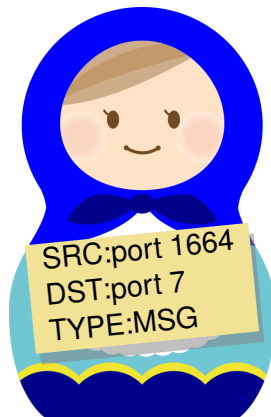
Exemple

Le module réseau lit les paramètres, et en déduit qu'il doit passer la poupée à la fonction UDP du module transport



Exemple

Le module transport lit les paramètres, et en déduit qu'il doit passer la poupée à la fonction messagerie du module application




Exemple

Le module application lit les paramètres, et voit que le message est pour Bob



Exemple

Bob lit le message



Bonjour!

- Les messages sont des suites d'octets
- Encapsuler un message consiste à ajouter, avant, ou après, certaines données

Exemple pratique (envoi)

La couche application de la machine A veut envoyer le message “bonjour” au serveur echo de la machine B.

626f6e6a6f7572

ECHO

bonjour

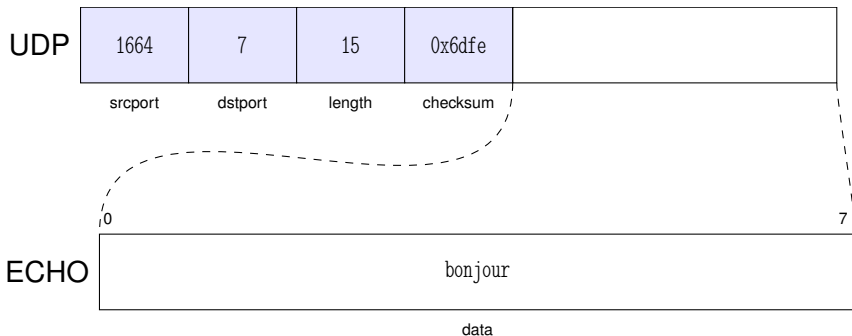
data

Exemple pratique (envoi)

Elle demande à sa couche transport (UDP), qui rajoute les champs nécessaires à la bonne compréhension du message

06800007000f

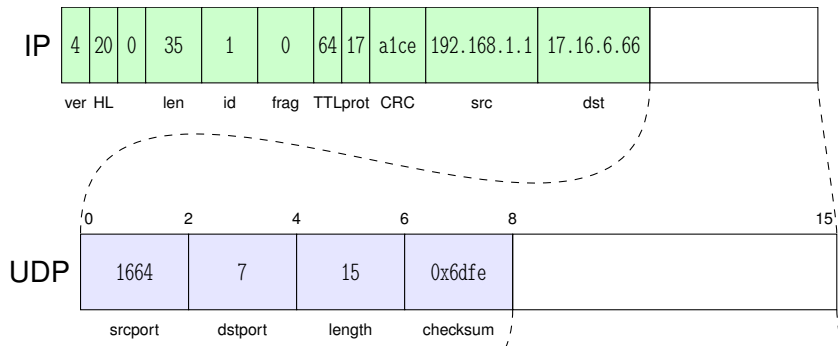
6dfe626f6e6a6f7572



Exemple pratique (envoi)

UDP demande à sa couche réseau (IP), qui rajoute les champs nécessaires à la bonne compréhension du message

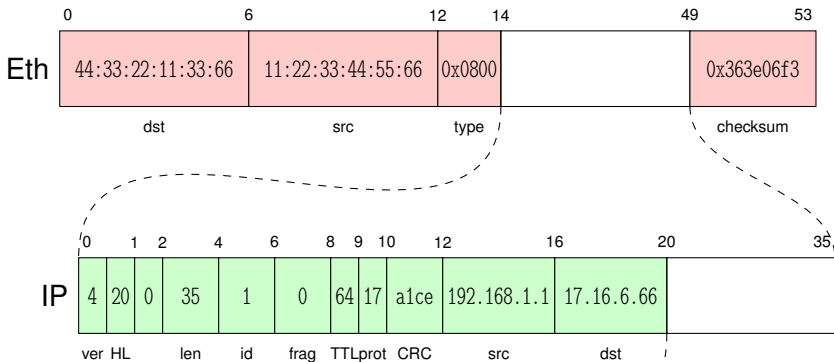
450000230001
00004011a1cec0a801011110064206800007000f
6dfe626f6e6a6f7572



Exemple pratique (envoi)

IP demande à sa couche liaison (Ethernet), qui rajoute les champs nécessaires à la bonne compréhension du message

4433221133661122334455660800450000230001
00004011a1cec0a801011110064206800007000f
6dfe626f6e6a6f7572363e06f3



Exemple pratique (envoi)

Ethernet demande à la couche physique d'envoyer le message

```
4433221133661122334455660800450000230001  
00004011a1cec0a801011110064206800007000f  
6dfe626f6e6a6f7572363e06f3
```

Exemple pratique (réception)

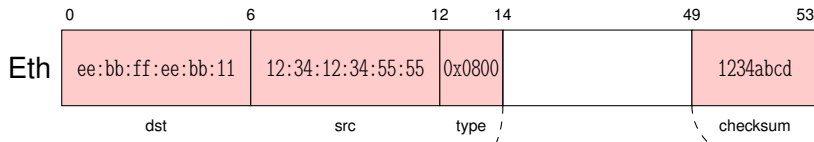
La couche physique reçoit le message suivant, transmis à la couche Ethernet

```
eebbffeebb111234123455550800450000230001  
00003d11a1cec0a801011110064206800007000f  
6dfe626f6e6a6f75721234abcd
```

Exemple pratique (réception)

Ethernet déchiffre le message et comprend qu'elle doit le donner au module IP (0x0800)

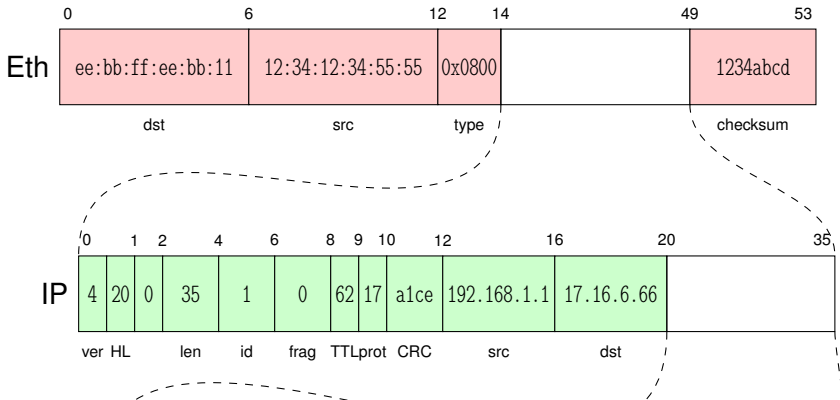
```
eebbffeebb111234123455550800450000230001  
00003d11a1cec0a801011110064206800007000f  
6dfe626f6e6a6f75721234abcd
```



Exemple pratique (réception)

IP déchiffre le message et comprend qu'elle doit le donner à UDP (0x11)

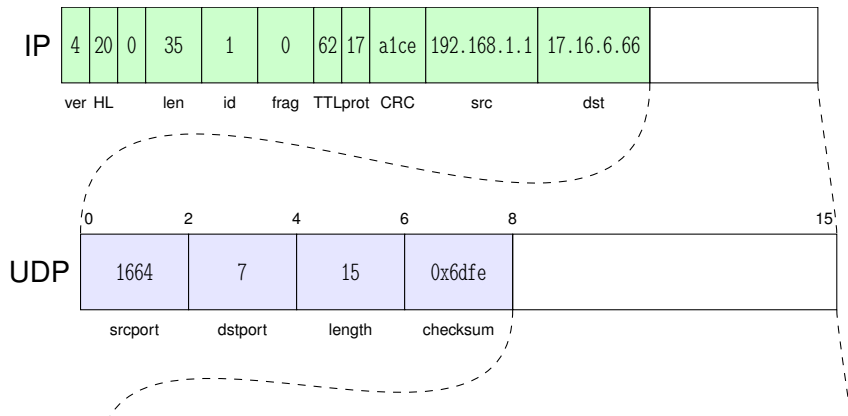
eebbffeebb11123412345550800450000230001
00003d11a1cec0a801011110064206800007000f
6dfe626f6e6a6f75721234abcd



Exemple pratique (réception)

UDP déchiffre et comprend que c'est pour l'application echo (7)

eebbffeebb111234123455550800450000230001
00003d11a1cec0a801011110064206800007000f
6dfe626f6e6a6f75721234abcd



Exemple pratique (réception)

L'application echo reçoit "bonjour"

```
eebbffeabb111234123455550800450000230001  
00003d11a1cec0a801011110064206800007000f  
6dfe626f6e6a6f75721234abcd
```

ECHO

bonjour

data

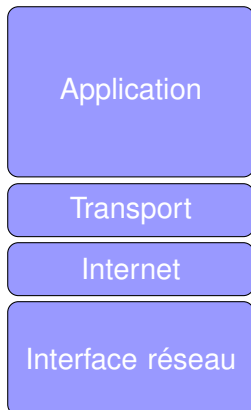
Le modèle...

... et la réalité

OSI



TCP/IP



Cours de réseaux = Apprentissage des différentes couches

Réseaux 1 :

- Couche physique
- Couche liaison
- Couche réseau

Réseaux 2 :

- Couche réseau
- Couche transport
- Couche application

Couche Physique

Physical Layer

Transmission d'une suite de bits d'un point *A* à un point *B* relié par un medium (cable, atmosphère)

- Potentiellement des erreurs de transmission (bruits) ou de décodage.

Couche Liaison

Data Link Layer

Transmission de *frames* d'un point *A* à un point *B* relié par un medium

- Sans erreurs
- Sans perte de messages
- Sans duplication

Couche Réseau

Network Layer

Transmission de données d'un point A à un point B en utilisant des points intermédiaires comme relai.

- Notion d'*adresse*, comprise par tous les points intermédiaires
- Notion de *routage*, indépendant ou non des données.
- Le message peut être perdu
- Divers bouts d'un message peuvent arriver dans un ordre différent

Couche Transport

Transport Layer

Transmission *fiable* de données entre un *processus A* et un processus *B* en utilisant des points intermédiaires comme relai.

- Multiplexage
- Fiabilité

- Couche session
 - Etablir une relation entre deux applications distantes souhaitant coopérer.
 - Reprise en cas d'incident majeur sur le réseau.
- Couche présentation
 - Représentation des données de façon universelle
 - Cryptographie et Compression des données.
- Couche application