

# Cours - Réseaux

Erwan DEMERVILLE, Cindy VAN HULLE, Théo BODDAERT

25 janvier 2023

# Table des matières I

- 1 Activité 1 : Adresses IP
- 2 Le modèle TCP/IP
- 3 Activité 2 : Découpage des paquets
- 4 Les protocoles TCP et UDP
- 5 Récupération de paquets perdus
- 6 TP Filius

## Section 1

### Activité 1 : Adresses IP

# 1) Une adresse = un identifiant

Une adresse IP est l'identifiant de la machine sur le réseau.

Les adresses IP doivent donc être uniques.

L'objectif de cette activité est de découvrir comment les adresses IP sont construites et utilisées :

- Imaginons que le lycée possède plusieurs bâtiments, dans lesquels il y a plusieurs étages, dans lesquels il y a plusieurs salles, dans lesquelles il y a plusieurs ordinateurs.
- Pour identifier un ordinateur de manière unique, nous allons nous servir de l'ensemble des identifiants :

Figure 1 : Plan satellite



Figure 1: Plan de l'Université

Figure 2 : Plan du bâtiment

Figure 3 : Plan de l'étage

Figure 4 : Plan de la classe



Figure 2: EPFL-ENAC | ENAC-IT | Les 2 salles de PCs d'enseignement  
ENAC-SSIE

L'ordinateur n°5 de la salle 10 n'a pas le même identifiant que l'ordinateur n°5 de la salle 11 puisque ce n'est pas le même ordinateur.

- a. Trouver un moyen pour construire chaque identifiant d'ordinateurs de manière unique.
- b. Chercher puis trouver l'identifiant de l'ordinateur sur laquelle vous êtes connecté.
- c. Noter des exemples d'identifiant des ordinateurs :
  - de vos voisins directs
  - de la salle voisine
  - de l'étage supérieur
  - du bâtiment voisin



## 2) Réseaux

Dessiner à la main à quoi le réseau du lycée pourrait ressembler.

## Section 2

### Le modèle TCP/IP

- Lorsque l'on souhaite transmettre une **lettre** à une personne, on place cette lettre dans une **enveloppe** sur laquelle on note l'**adresse du destinataire**. Cette lettre est placée dans la **boîte au lettre la plus proche**, puis elle sera acheminée vers un **centre de tri**, puis parcourra plusieurs autres centres de tri avant d'arriver à **destination**.
- Sur **Internet**, le principe est similaire. La donnée que l'on souhaite transmettre d'une **machine** à une autre va être **enveloppée** par différents **protocoles** qui vont y ajouter de l'**information**. Ce procédé s'appelle l'**encapsulation**.

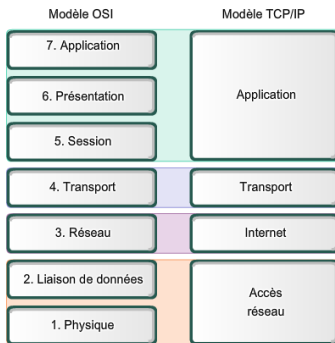
# Vous avez dit “protocole” ?

- Un **protocole** est un **ensemble de règles** à respecter pour qu’une **couche** d’un système puisse communiquer avec la même couche du système cible.
- Les protocoles permettent de **standardiser** la communication entre équipements.
- Pour reprendre l’exemple précédent d’une **transmission d’une lettre par la poste**, les règles seraient les suivantes :
  - Placer la lettre dans une enveloppe,
  - Sur le recto de l’enveloppe, écrire l’adresse du destinataire,
  - Sur le verso, écrire l’adresse de l’expéditeur.

# Un modèle en couche

- Pour décrire comment les systèmes réseau communiquent et envoient des **données** d'un **expéditeur** à un **destinataire**, un **modèle en couches** a été proposé. Il s'agit de décrire les réseaux sous la forme d'un ensemble de **couches superposées** les unes aux autres.
- Le **modèle OSI** est un modèle en **7 couches** proposant un **cadre général conceptuel** pour la création de normes cohérentes.
- Un **modèle** plus proche de la réalité est le **modèle TCP/IP**, en **4 couches**. C'est ce modèle-là que l'on abordera par la suite.

## Comparaison des modèles OSI et TCP/IP



Les principaux parallèles concernent les couches transport et réseau.

Figure 3: Modèle OSI - TCP/IP

Source : <http://tomwiki.wdfiles.com/local--files/wiki%3Aosi-et-tcpip/OSI%20Vs%20TCP%20IP.png>

- Dans un tel modèle, l'étude du « tout » est réduit à celle de ses « parties », cela rend l'ensemble **plus facile à manipuler**.
- Les couches sont **relativement indépendantes les unes des autres**. Plus l'on descend dans les couches, plus on se rapproche du niveau machine, en commençant par la **couche application** (plus *haut niveau*), et en finissant par la **couche physique** (plus *bas niveau*).
- En descendant dans les différentes couches, l'information à transmettre est **segmentée** (on parlera de *segments/datagrammes*, de *paquets*, de *trames*), chaque couche ajoutant de l'information (principe de l'**encapsulation**).

# Description des différentes couches

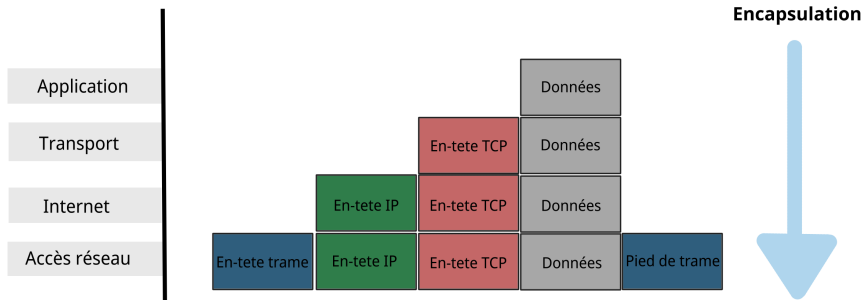


Figure 4: Schéma d'encapsulation



## La couche Accès réseau

- Cette couche correspond aux couches **Physique** et **Liaison des données** du modèle **OSI**.
- Elle décrit les **caractéristiques physiques** de la communication et assure le **transfert fiable** de **blocs d'octets** appelés **trames Ethernet** entre systèmes raccordés au même médium (câble, Wi-Fi, Ethernet...)
- Parmi les informations d'**en-tête** ajoutées aux données reçues depuis la couche supérieur, elle indique les **adresses MAC de la source et du destinataire**.

## La couche Internet

- Son rôle est d'assurer le **transfert de blocs d'octets** entre deux systèmes distants pas forcément raccordés au même médium.
- Elle assure l'envoi de **paquets** avec le **protocole IP** dans n'importe quel réseau sans qu'**aucun chemin** pour le transfert des données ne soit **établi** à l'avance (protocole « non orienté connexion »).
- Les paquets émis peuvent suivre des **routes différentes** et **arriver dans le désordre**. C'est la **couche Transport** qui se chargera de remettre les paquets dans le bon ordre.
- Parmi les informations d'**en-tête** ajoutées aux données reçues depuis la couche supérieur, elle indique les **adresses IP de la source et du destinataire**.

## La couche Transport

- Son rôle est similaire à celui de la couche **Transport** du modèle **OSI**.
- Elle utilise les protocoles **TCP** et **UDP**.
- **UDP** est moins fiable **mais plus rapide** que **TCP**. Il est utilisé par exemple pour le *streaming* ou dans les liaisons *voix IP*, où la perte de quelques données est moins grave que la limitation de la vitesse.
- Dans l'**en-tête TCP**, on retrouve le **port source** et le **port de destination** (80 pour le protocole HTTP, ou 21 pour le FTP par exemple).

## La couche Application

- Elle assure l'**échange** et la **mise en forme** de **données d'application** (selon l'application).
- Elle regroupe tous les protocoles de haut niveau (FTP, HTTP, HTTPS, DHCP, DNS, SSH, POP, SMTP...). Elle effectue également le choix du protocole de transport (**TCP** ou **UDP**) **adapté au service** demandé.

## Section 3

### Activité 2 : Découpage des paquets

## Activité 2 : Découpage des paquets

Lorsqu'on envoie des données à une machine  $B$  depuis une machine  $A$ , il arrive parfois que ces données soient trop grandes pour pouvoir être envoyées d'un coup. Pour faire l'analogie, une voiture sert au transport sur un réseau routier. On ne peut pas placer un lit King Size dans une Twingo !

La solution, en réseau, c'est de découper les données en "paquets" :

- a.
  - Ecrire sur un papier le message `hello world`. Imaginons que l'envoi de ce message n'est pas possible en une seule fois.
  - Découper le message, caractère par caractère avec une paire de ciseaux. Chaque bout de papier est un paquet. Seulement, les paquets n'utilisent pas nécessairement la même route pour atteindre la destination. Il se peut que certains paquets utilisent des routes plus longues.
- b. Pour éviter de recevoir le message dans le désordre, à chaque paquet, inscrire le numéro du paquet.
- c. Envoyer les paquets un par un dans le réseau de la classe.
- d. Après avoir reçu un message, remettre dans l'ordre des numéros de paquets les paquets.

## Section 4

# Les protocoles TCP et UDP



- Le protocole **UDP** permet de **définir des ports** permettant de déterminer **quels logiciels doivent réceptionner chaque paquet**. Les données sont transmises sous forme de **datagrammes**. Toutefois, des paquets peuvent se perdre, et rien ne permet de s'assurer de leur bonne réception.
- Aucune communication préalable n'est requise pour établir la connexion. On dit que le protocole **UDP** est un mode de transmission sans connexion.
- Il est utilisé quand il est nécessaire soit de transmettre des données très rapidement, et où la perte d'une partie de ces données n'a pas grande importance (par exemple la video).

- Le protocole **TCP** (Transmission Control Protocol) est un **ensemble de règles** qui fixe les conditions de transport des informations. Il est **plus fiable** et assure **la bonne réception de tous les paquets**. Une **connexion** est établie entre deux entités à l'aide d'un processus de **handshaking**. Les données sont découpées en **segments** de taille définie par la **MTU** (Maximum Transmission Unit) du réseau sous-jacent.
- Le protocole **TCP** est souvent constitué de trois phases:
  - *Établissement d'une connexion*: Avant d'envoyer des informations ou de demander des informations, une synchronisation entre l'émetteur et le receveur est assurée via un échange préalable.
  - *Transfert de données*: Pendant cette phase, plusieurs processus sont en action. Tout d'abord, si le paquet d'envoi est trop gros, il est fractionné et les paquets sont numérotés. Les paquets sont envoyés et un accusé de réception est attendu pour chaque paquet envoyé. Si un accusé de réception n'est pas reçu, le paquet est réexpédié.
  - *La rupture de la liaison*: Lorsque l'ensemble des transferts sont effectués, on procède à l'arrêt de la connexion.

Le principe du **handshaking** peut être représenté de la manière suivante :

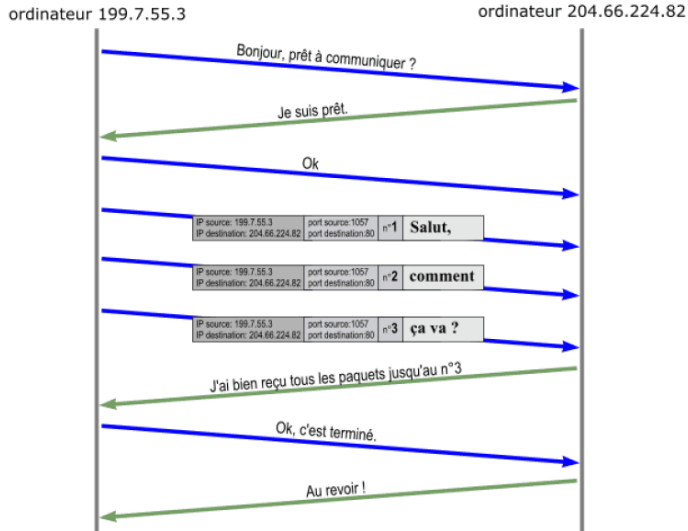


Figure 5: Processus du handshaking

## Section 5

### Récupération de paquets perdus

# Protocole de bit alterné

- Nous avons vu que le protocole TCP propose un mécanisme d'accusé de réception afin de s'assurer qu'un paquet est bien arrivé à destination. On parle plus généralement de processus d'acquittement. Ces processus d'acquittement permettent de détecter les pertes de paquets au sein d'un réseau, l'idée étant qu'en cas de perte, l'émetteur du paquet renvoie le paquet perdu au destinataire. Nous allons ici étudier un protocole simple de récupération de perte de paquet : le **protocole de bit alterné**.
- Le principe de ce protocole est simple, considérons 2 ordinateurs en réseau : un ordinateur A qui sera l'émetteur des trames et un ordinateur B qui sera le destinataire des trames. Au moment d'émettre une trame, A va ajouter à cette trame un bit (1 ou 0) appelé drapeau (flag en anglais). B va envoyer un accusé de réception (acknowledge en anglais souvent noté ACK) à destination de A dès qu'il a reçu une trame en provenance de A. À cet accusé de réception on associe aussi un bit drapeau (1 ou 0).

- La règle est relativement simple : la première trame envoyée par A aura pour drapeau 0, dès cette trame reçue par B, ce dernier va envoyer un accusé de réception avec le drapeau 1 (ce 1 signifie “la prochaine trame que A va m’envoyer devra avoir son drapeau à 1”). Dès que A reçoit l’accusé de réception avec le drapeau à 1, il envoie la 2e trame avec un drapeau à 1, et ainsi de suite...

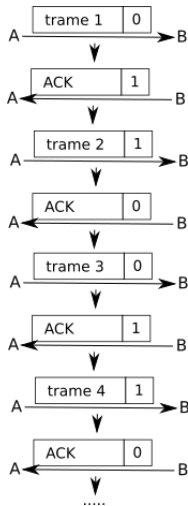


Figure 6: Protocole du bit alterné

- Le système de drapeau est complété avec un système d'horloge côté émetteur. Un "chronomètre" est déclenché à chaque envoi de trame, si au bout d'un certain temps, l'émetteur n'a pas reçu un acquittement correct (avec le bon drapeau en accusé de réception), la trame précédemment envoyée par l'émetteur est considérée comme perdue et est de nouveau envoyée.



## Section 6

### TP Filius

...