

# Chap 1 - Introduction

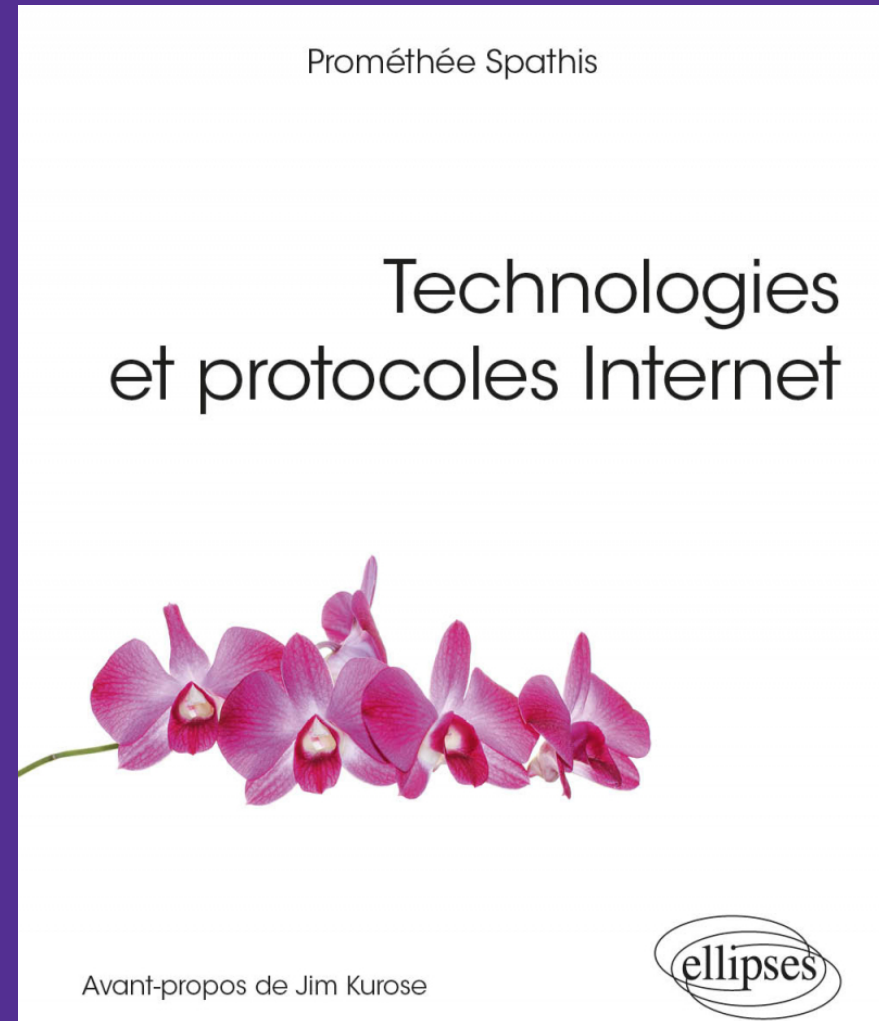
Ces transparents sont mis à disposition de tous (étudiants, enseignants, lecteurs).

En contrepartie, merci de bien vouloir :

- mentionner leur source,
- préciser la mention de copyright.

Merci et bon cours !

© 2020 - 2023 Promethee Spathis  
All Rights Reserved



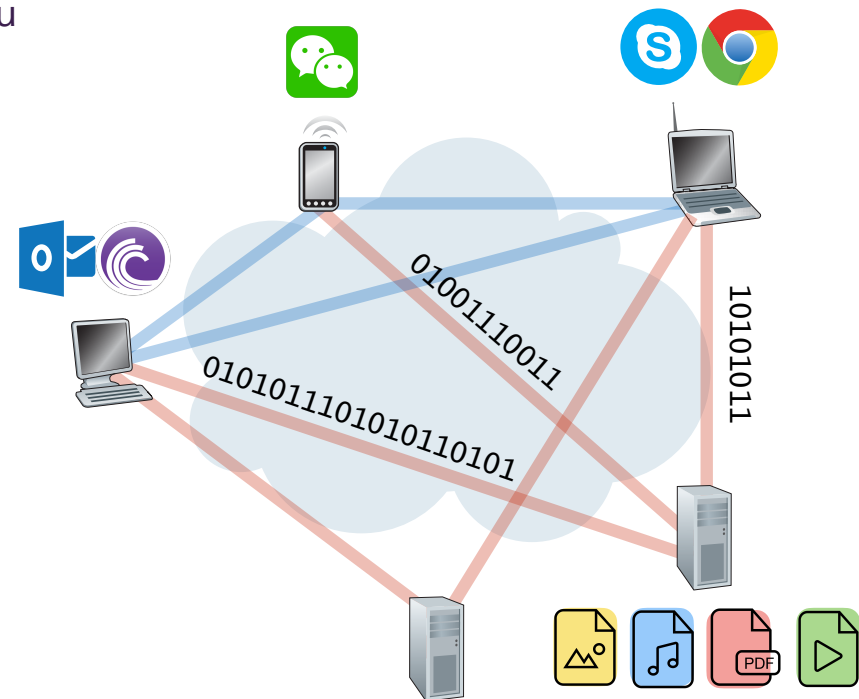
# Plan du cours

- Les réseaux de données
  - pour qui et pour quoi faire ?
- Les applications, à l'origine des données
  - Clients et serveurs
- Empaquetage des données
  - encapsulation et désencapsulation
- Architectures en couches
  - couches et protocoles
- Topologie des réseaux
  - types de nœuds et répartition des tâches

# Réseaux de données

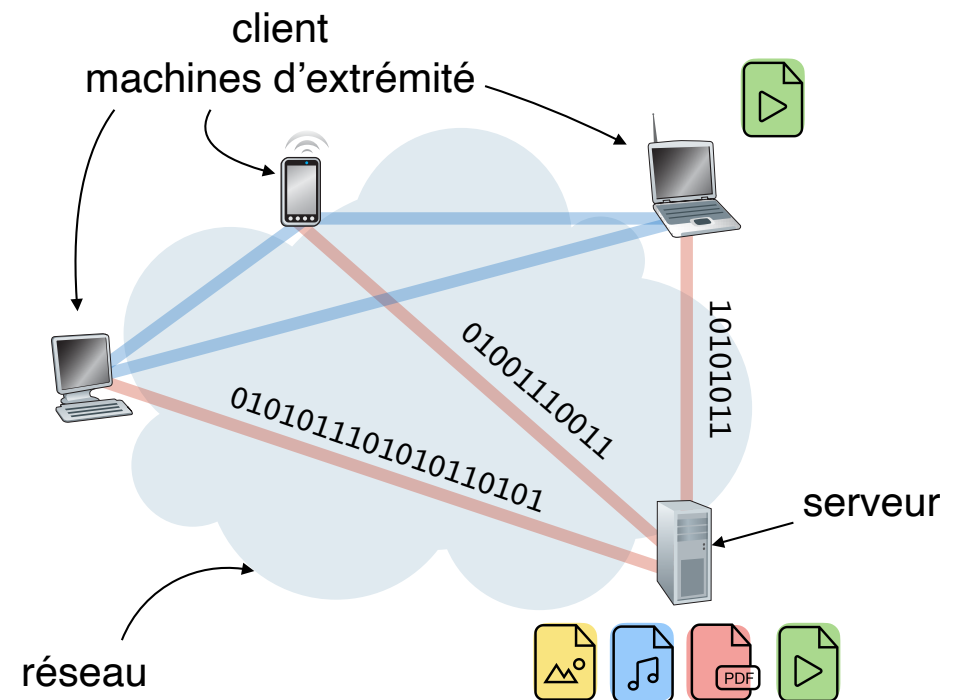
## Que sont-ils et à quoi servent-ils ?

- Les réseaux de données connectent des machines
  - ordinateurs : processeur, mémoire, cartes réseau
  - machines de bureau, portables, tablettes, smartphones, 'objets' (IoT)
- Les utilisateurs veulent accéder à des ressources partagées
  - contenus (pages web, images, vidéos, ...),
  - services (stockage, traitement, ...)
- Les réseaux de données transportent les données pour le compte des utilisateurs
  - Les utilisateurs Internet téléchargent et installent des applications sur leur machine
  - Les opérateurs de réseau peuvent restreindre l'utilisations de certaines applications (e.g., violation des droits d'auteur)



# Les données : d'où viennent-elles ?

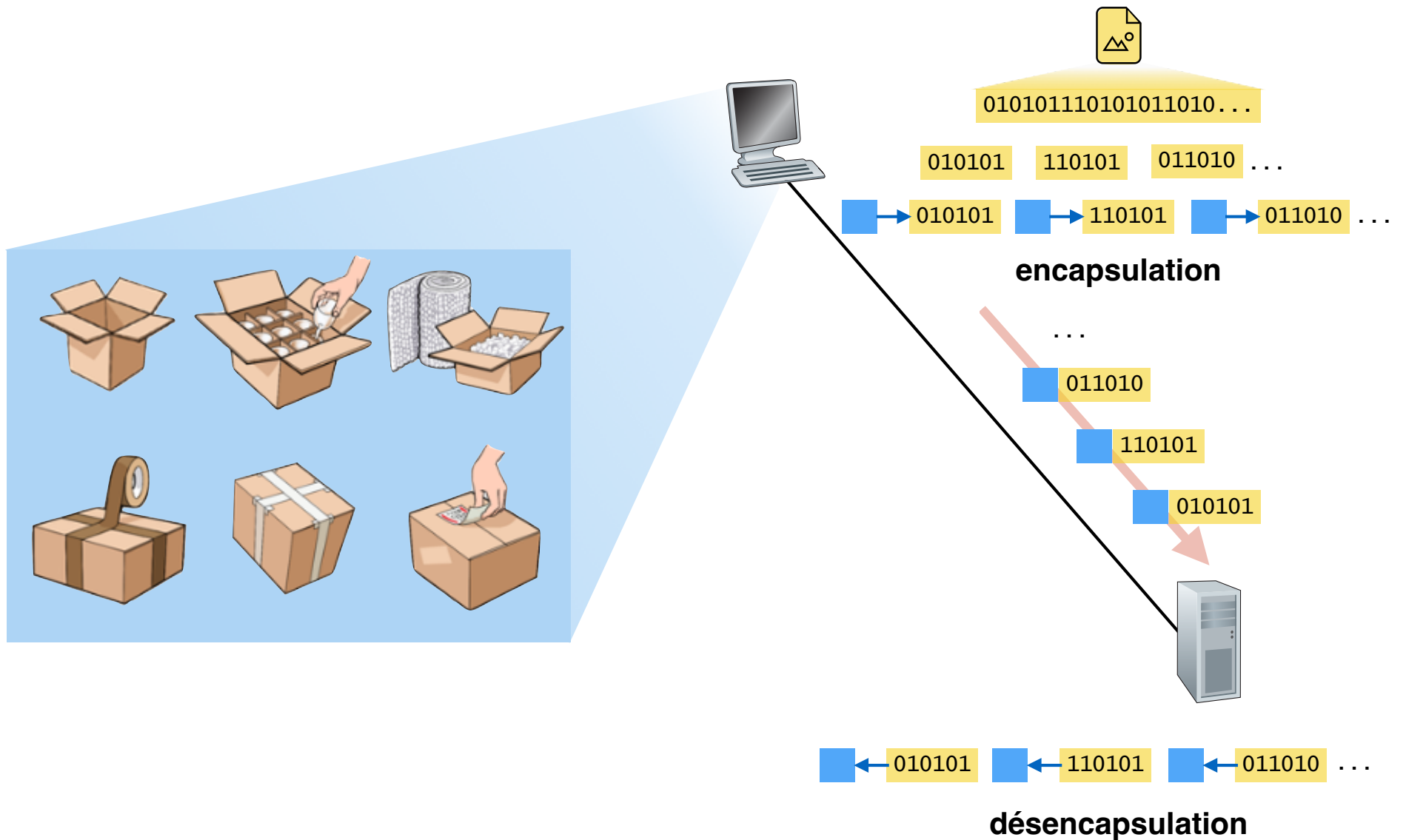
- Un réseau de données connecte des machines
  - les applications sont des logiciels qui exécutent des tâches spécialisées pour le compte des utilisateurs
  - les machines d'extrémité hébergent les applications réseau
- Les applications réseau font référence à (au moins) deux logiciels
  - chaque logiciel tourne sur une machine distante
  - ces logiciels produisent et consomment des bits de données
- Les applications réseau sont conçues selon deux modèles
  - asymétrique: Client/Serveur
  - symétrique: Pair-à-Pair (P2P)



# Les applications réseau

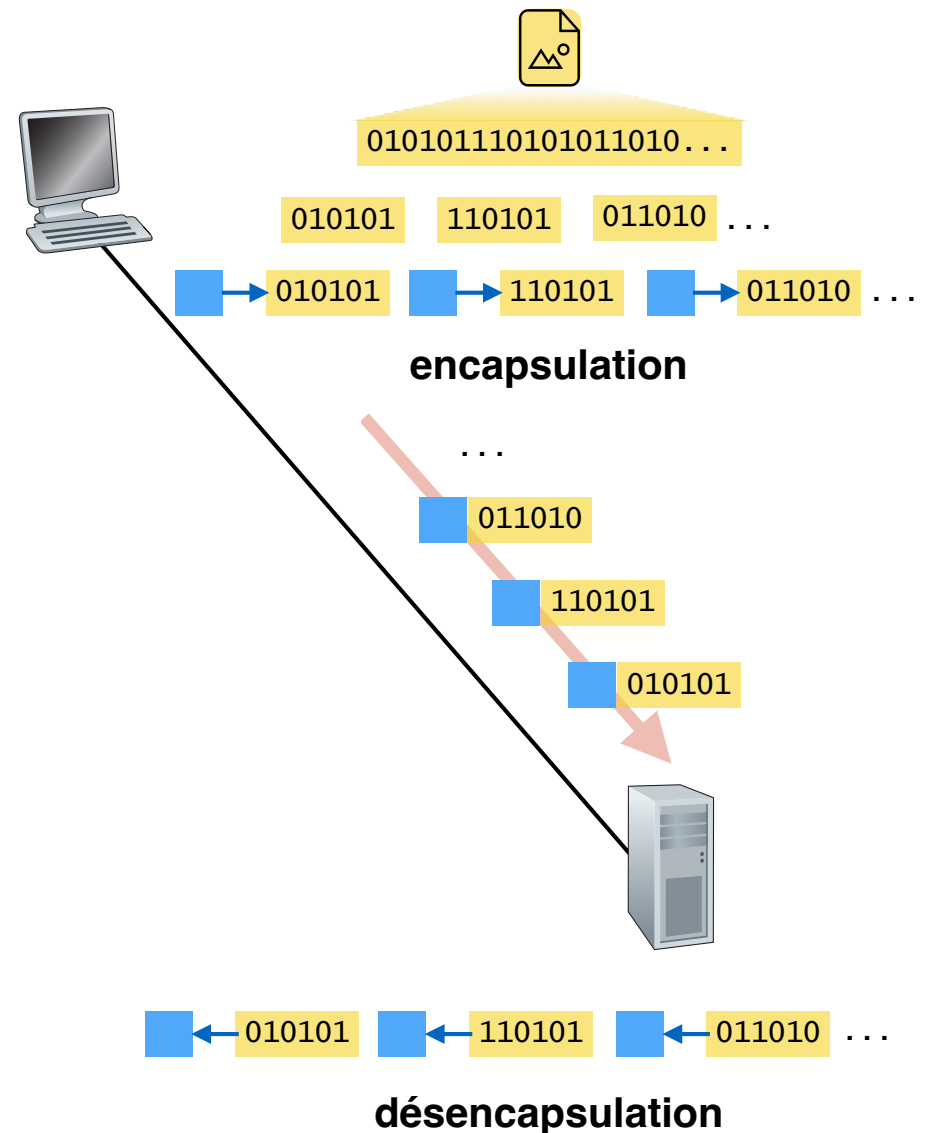
- Les machines d'extrémité exécutent des applications réseau
  - hébergées à la périphérie des réseaux
  - les machines impliquées communiquent par l'envoi de requêtes et la réception des réponses correspondantes
- Conception asymétrique : modèle Client/Serveur
  - le programme côté client diffère de celui côté serveur
  - les clients soumettent des requêtes (e.g. demandes de téléchargement ou d'hébergement de fichiers)
  - un serveur traite les requêtes client (e.g. par l'envoi ou le stockage d'un fichier)
- Conception symétrique : modèle Pair-à-Pair
  - tous les noeuds exécutent le même programme
  - les noeuds sont des machines d'extrémité qui agissent tour à tour comme client ou serveur
  - e.g. applications P2P de partage de fichiers :
    - Napster, Gnutella, LimeWire, BearShare, BitTorrent, ...

# Emballage des données



# Emballage des données

- Les gros fichiers sont découpés en 'petits' morceaux de données
  - chaque morceau a une taille maximale et minimale
- Des bits supplémentaires sont ajoutés à chaque morceau de données
  - ajoutés en tête des morceaux de données
  - contiennent des informations de contrôle (i.e., adresses source et destination, numéro du morceau, ...)
  - l'entête et les bits de données forment un message
- L'entête est ajouté par la source, retiré par la destination
  - ajout d'un entête : encapsulation
  - retrait de l'entête : désencapsulation
- Plusieurs entêtes sont nécessaires
  - afin de remplir les propriétés attendues concernant le transfert de données

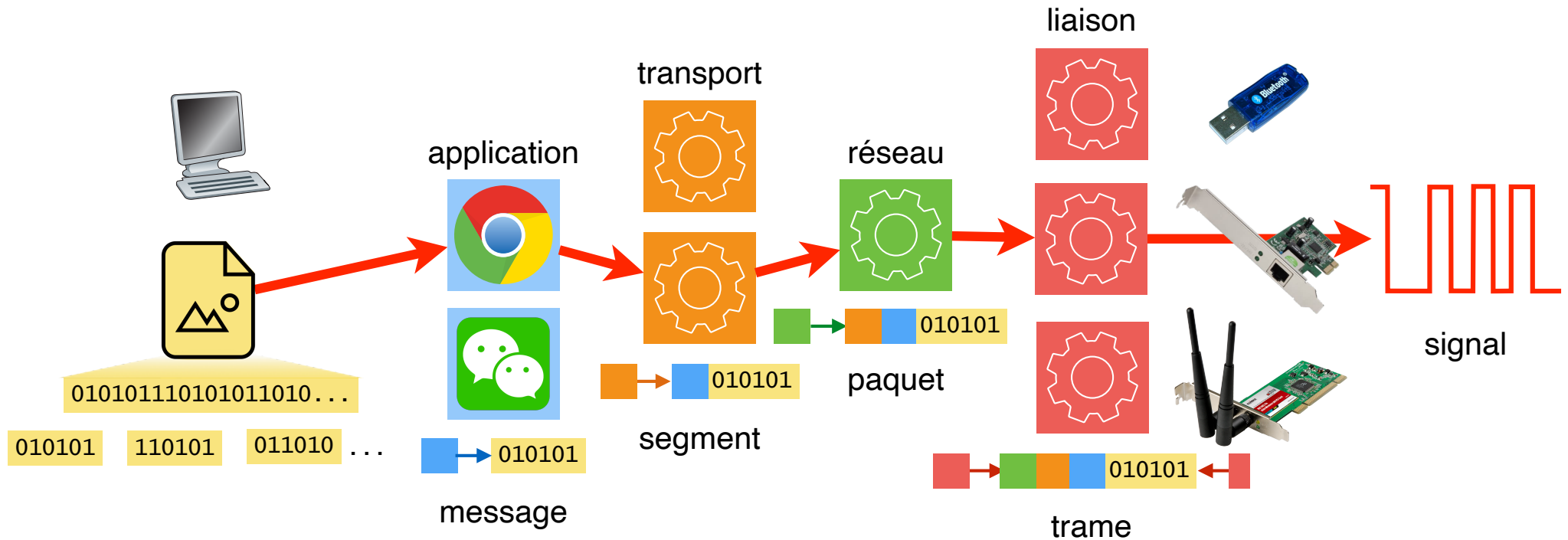


# Propriétés des transferts de données

- **Fiabilité: réparation des erreurs et des pertes**
  - l'entête contient des bits de type 'papier bulle'
    - bits de détection/correction d'erreur
    - numéro de séquence des messages
    - ...
- **Efficacité: maximisation de l'utilisation des ressources**
  - remplir les messages avec des données suffisantes pour atteindre la taille max des messages
  - envoyer les messages en nombre suffisant (tout en évitant la congestion)
- **Equité**
  - allocation équitable des ressources réseau (e.g., à quelle fréquence envoyer les messages)
  - neutralité des réseaux
- **Facteur d'échelle**
  - les performances sont peu voir non affectées par l'accroissement d'une grandeur mesurable
    - nombre d'utilisateurs, de nœuds, de messages, ...
- **Livraison ponctuelle**
  - les messages sont reçus à des instants réguliers dans le temps
- **Sécurité**
  - cryptage, confidentialité, authentification, ...
- ...

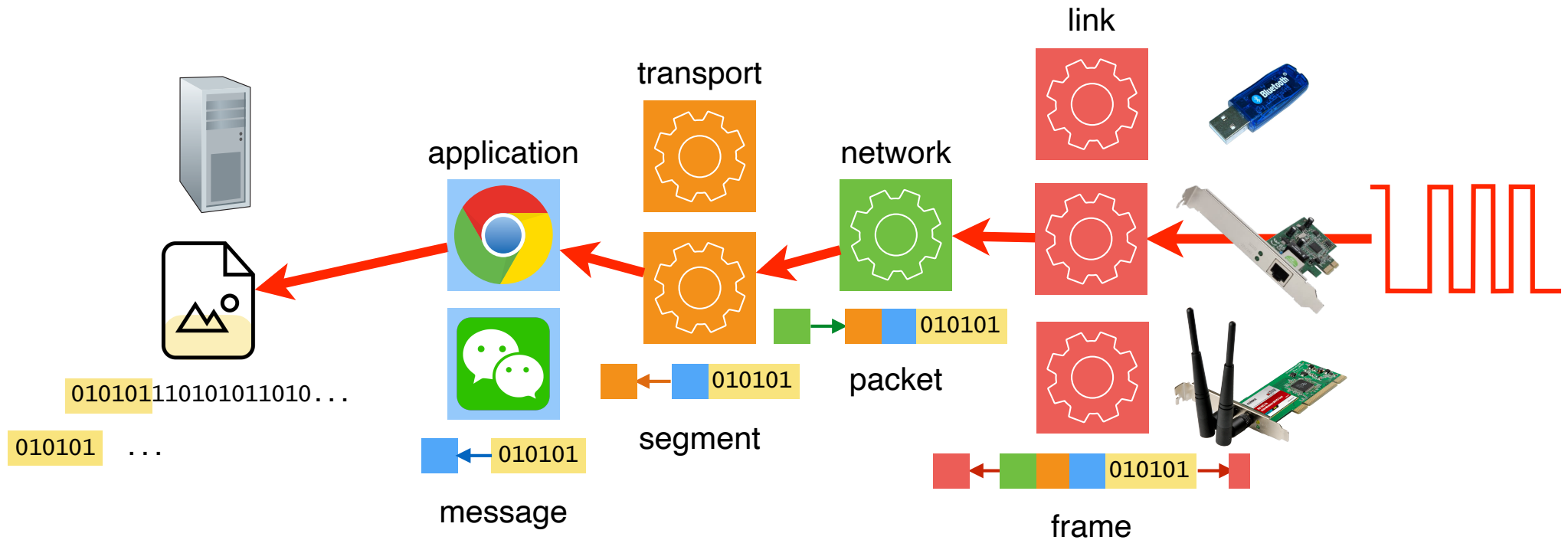


# Encapsulation des données



- Avant de quitter votre ordinateur, les bits de données traversent une séquence de programmes
- Chaque programme reçoit une séquence de bits en entrée
  - division de la séquence en fragments plus petits
  - ajout d'une entête (le dernier programme ajoute une queue)
    - entêtes et queue contiennent une liste spécifique de champs
  - installation et mise à jour d'états (un état est une valeur stockée en mémoire i.e. une variable)
  - le message en sortie d'un programme est passé au programme suivant
    - le dernier programme produit un signal électromagnétique transmis sur le support de transmission

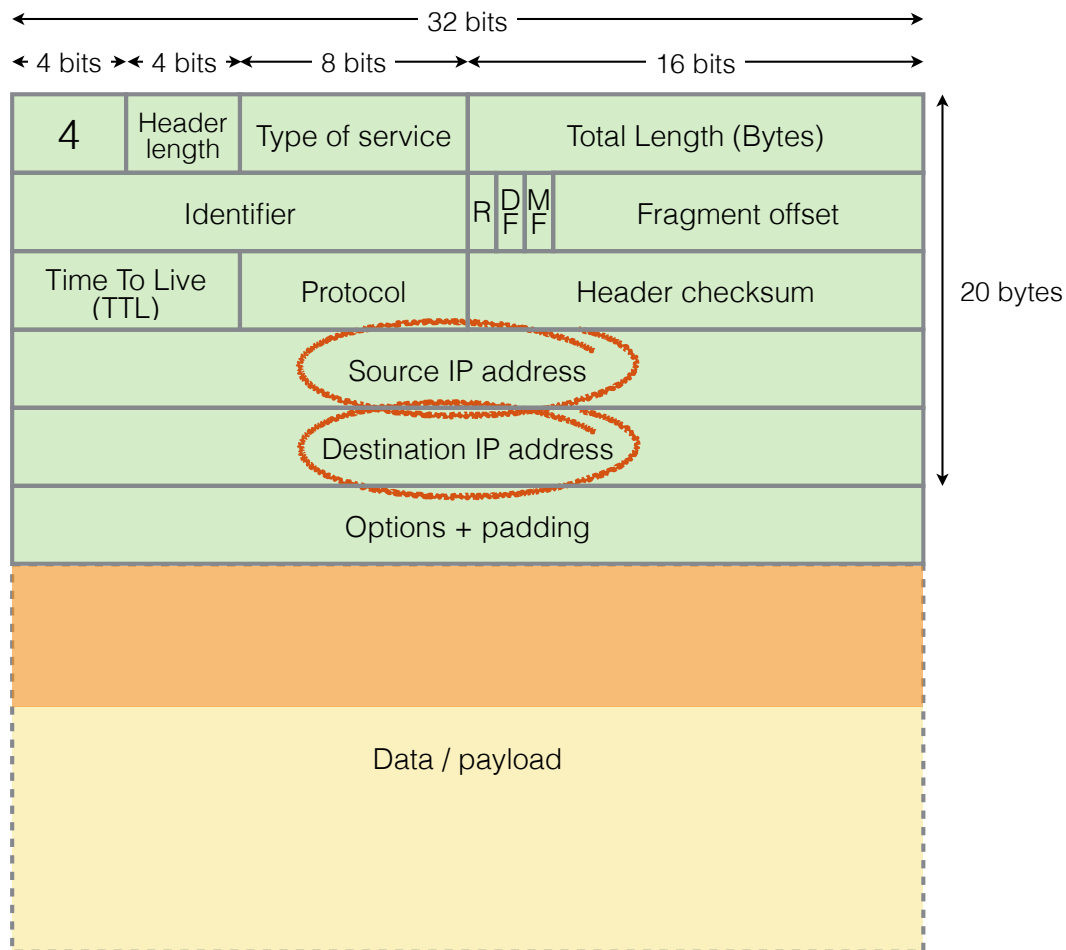
# Désencapsulation des données



- Côté récepteur, chaque programme :
  - retire l'entête et l'enqueue (si applicable)
  - installe ou met à jour des états
  - réassemble plusieurs fragments pour reconstruire le message d'origine
  - répond en retournant un message spécifique
  - passe le message désencapsulé (sans entête ni enqueue) et réassemblé au programme suivant

# Entête de paquet

## Entête Réseau (Paquet IP)



## Etiquette d'expédition FedEx

**FedEx Ground** Customer Receipt:  
**Call Tag Number**  
**9970123 00001234**

**From:**  
Jane Doe  
The Sample Company  
123456 Main St.  
Detroit, MI 48223  
800-123-1234

**Work Area:** Stan  
**Confirmation #** 997012300001234  
**Print Date:** 09/26/14 **Reprint #** 2  
**Scheduled Date:** 09/30/14  
**Contents:** NA

**TO:**  
XYZ PRINT SUPPLY CO.  
Attn: Tom Smith  
1234 SOUTH STREET  
KALAMAZOO, MI 49002

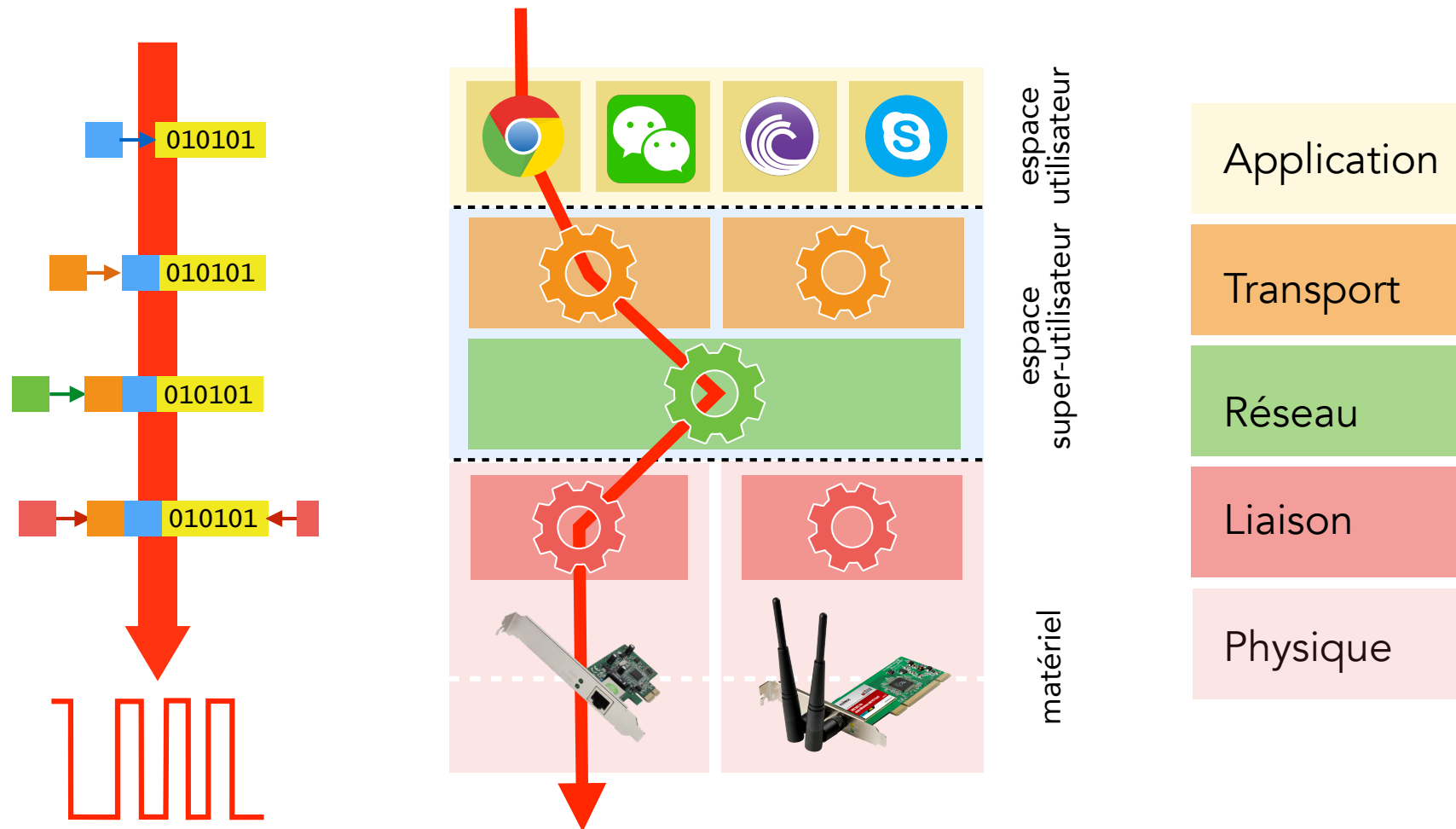
**FedEx Ground**  
**G**

**Ret#:**  
**RMA#:**

**S#:** 1234567  
**1 of 1**

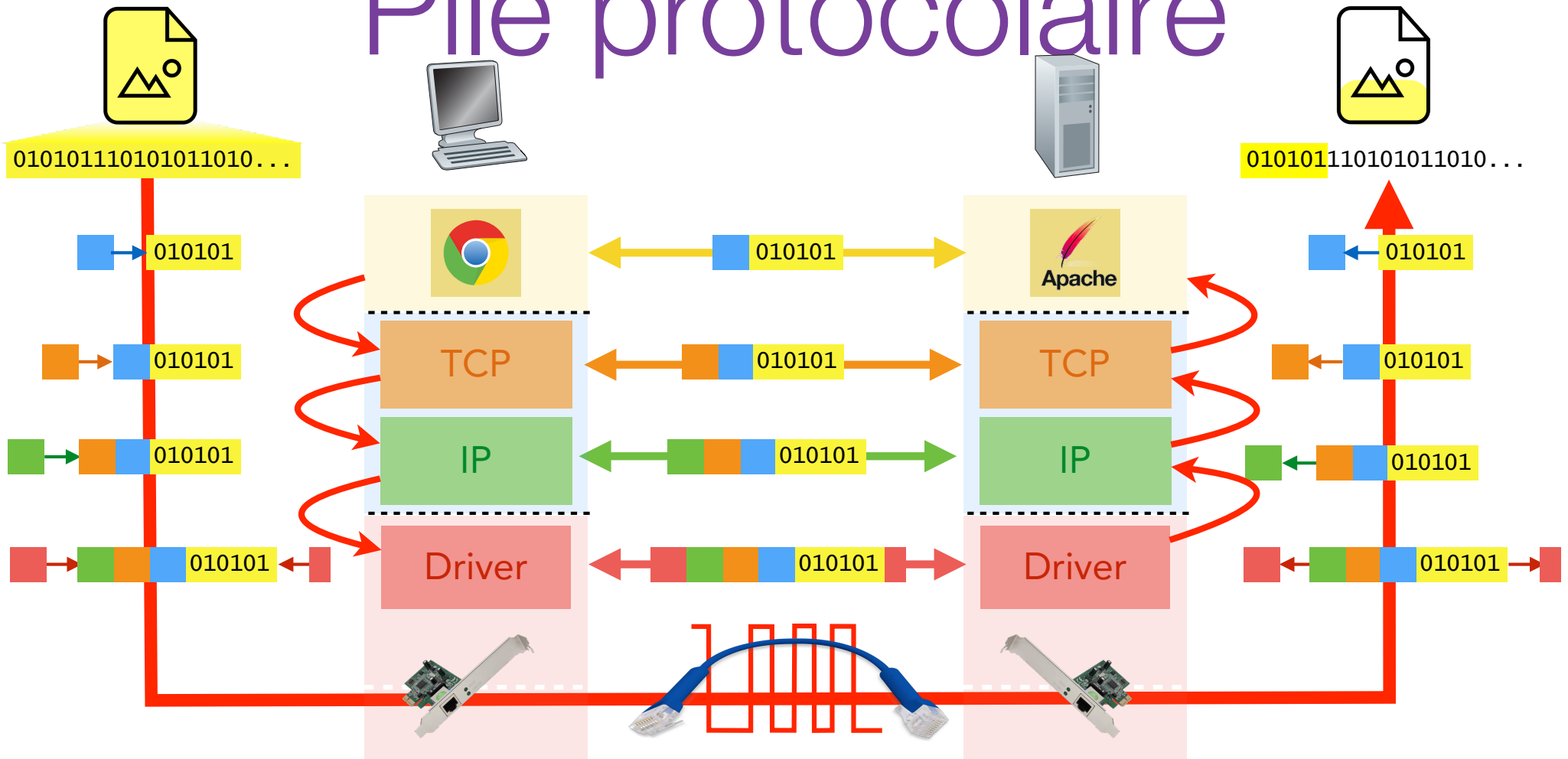
**(9612143) 1234567 12345678**

# Architecture en couches



- Par soucis de représentation, les programmes sont empilés
  - chaque programme appartient à une couche
- Chaque couche implémente plusieurs programmes selon le service attendu

# Pile protocolaire

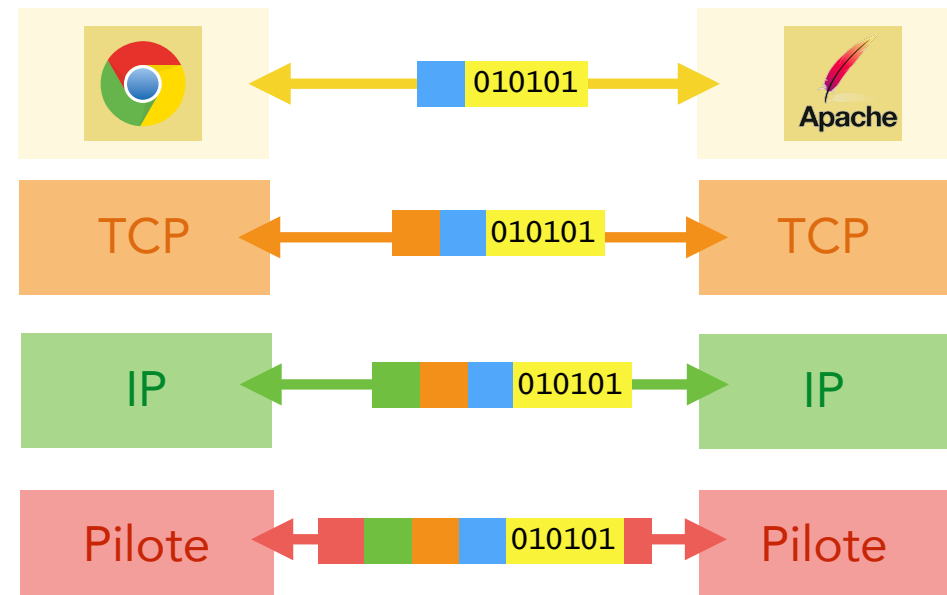


- Le programme d'une couche interagit avec son programme homologue logé sur une machine distante
- Programmes homologues interagissent en échangeant différents types de message selon des règles pré-établies
  - ces règles et types de messages sont définis au travers d'un protocole

# Protocoles

- Sources et destinations utilisent des programmes homologues pour communiquer
- Le format des messages échangés et les actions à exécuter à l'envoi ou la réception d'un message est spécifié dans un protocole

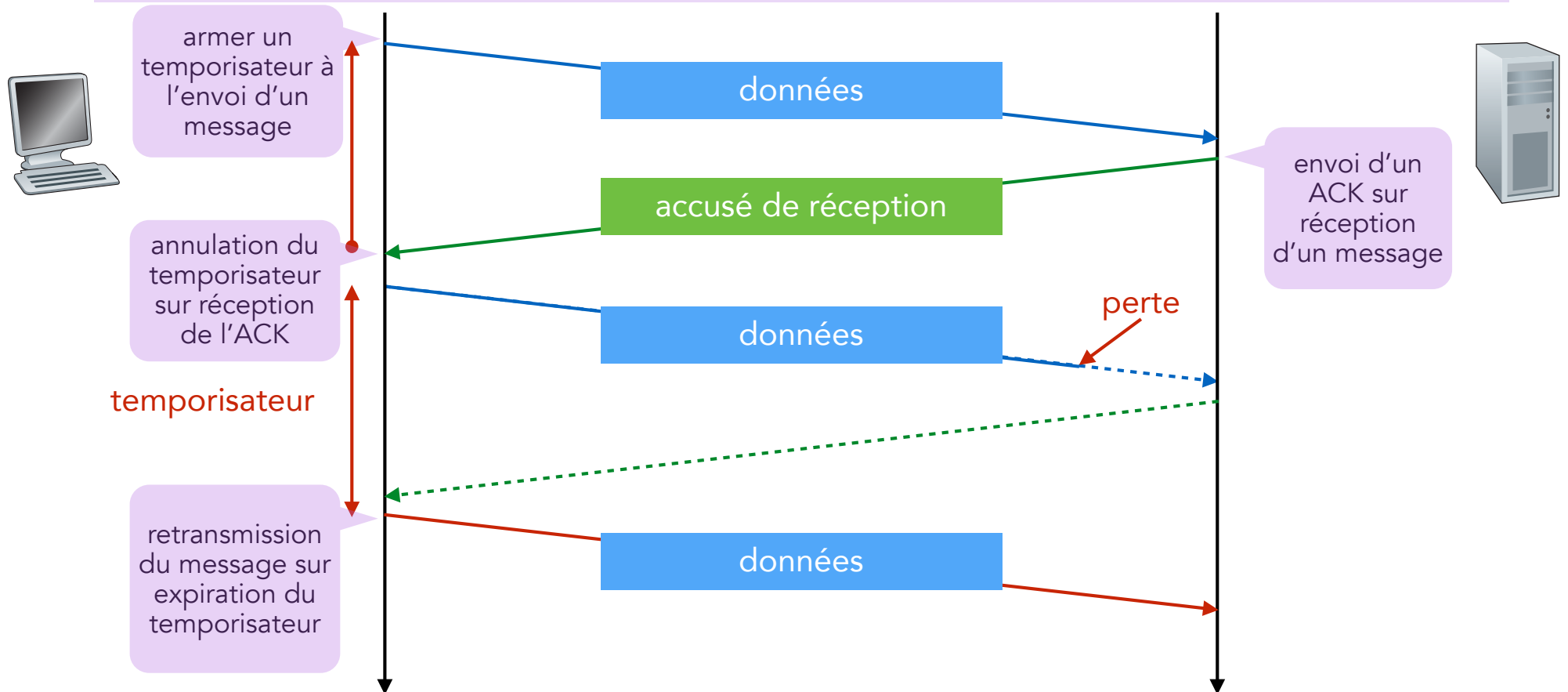
- Un protocole définit :
  - différent types de messages (i.e. des unités de données protocolaires)
    - un message consiste
      - d'une entête (et d'une enqueue)
      - de données (éventuellement)
    - l'entête (et l'enqueue)
      - contient une suite de champs
      - cette suite varie selon le type de message
  - un ensemble de règles
    - que faire avant/après l'envoi ou la réception d'un message
    - quels états installer, mettre à jour, supprimer



- Une communication résulte de l'exécution de plusieurs protocoles
  - chaque protocole appartient à une couche distincte

# Protocoles : les services rendus

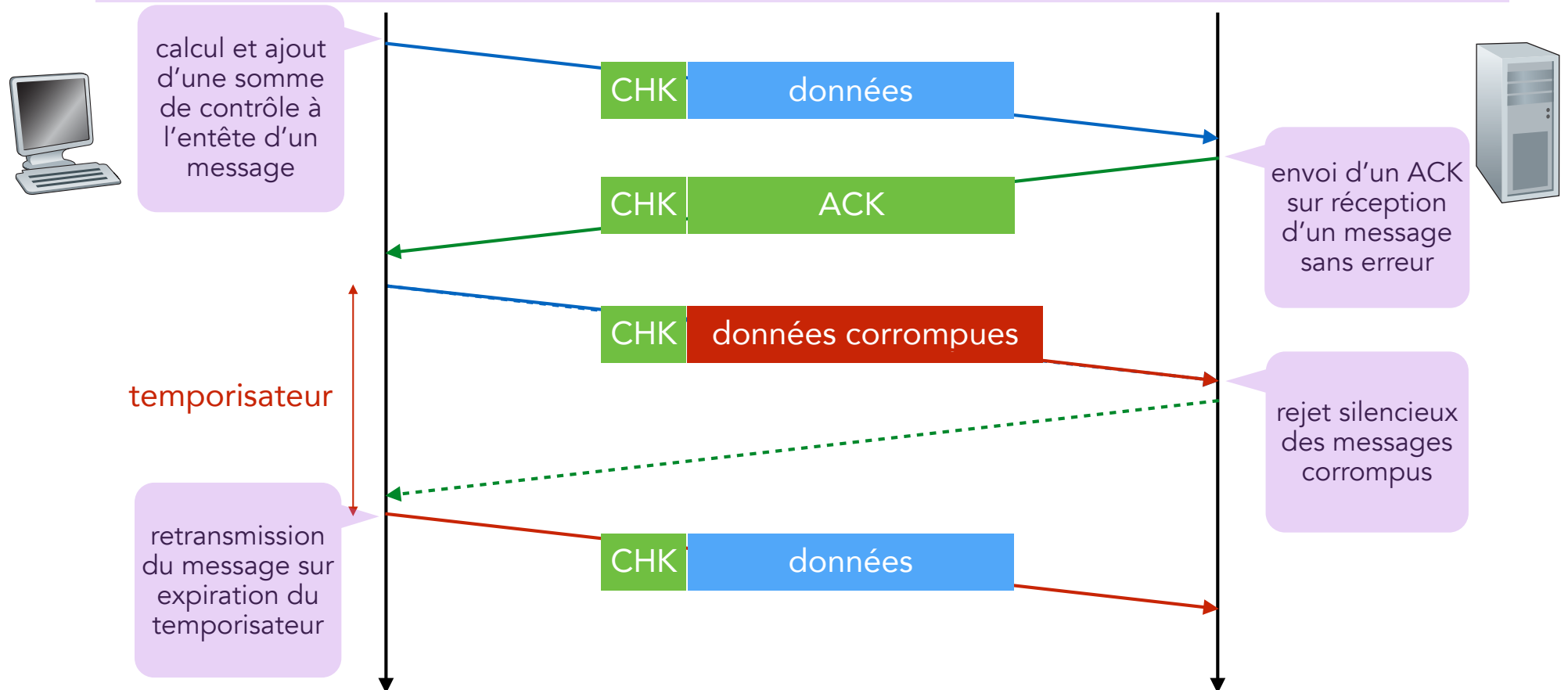
## Détection et réparation des pertes



- A l'envoi d'un message, la source enclenche un temporisateur
- Si le temporisateur expire avant la réception d'un ACK, le message est retransmis

# Protocoles : les services rendus

## Détection et correction d'erreur

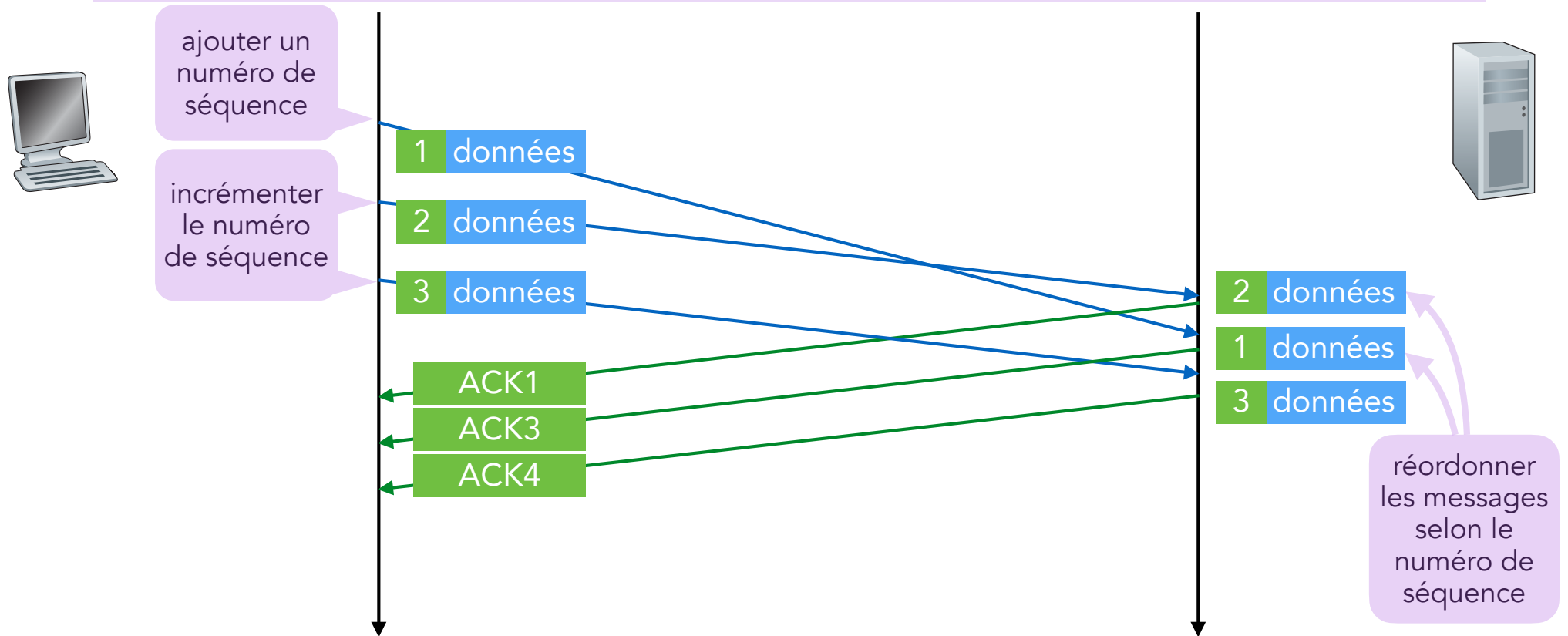


- La source calcule et ajoute des bits de contrôle à l'entête des messages
- Si les données reçues et les bits de contrôle ne sont pas conformes, le récepteur ignore le message



# Protocoles : les services rendus

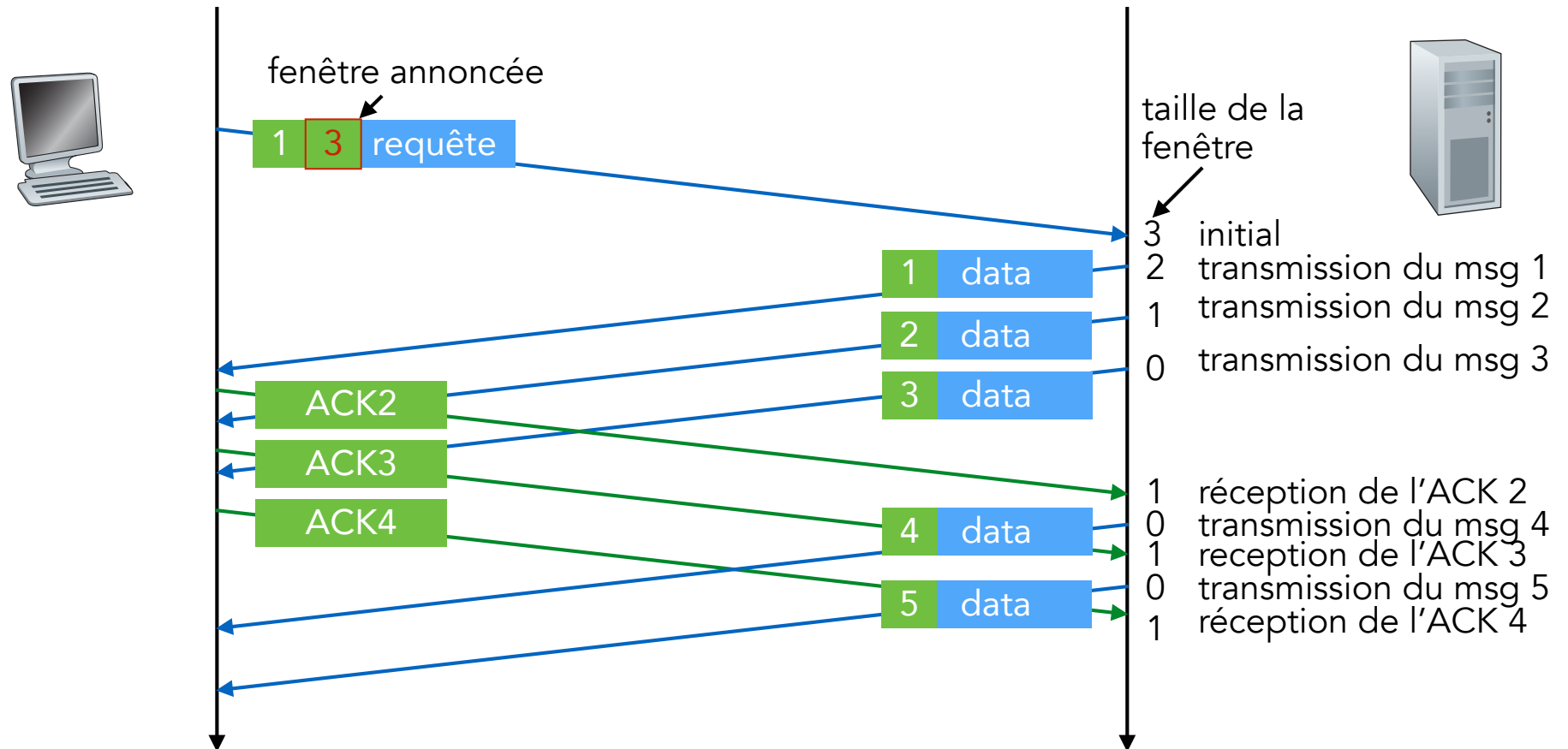
## Remise en séquence



- Un numéro de séquence est ajouté dans l'entête des messages
- Les ACK annoncent le numéro de séquence du prochain message attendu
- Le récepteur réordonne les messages reçus en désordre

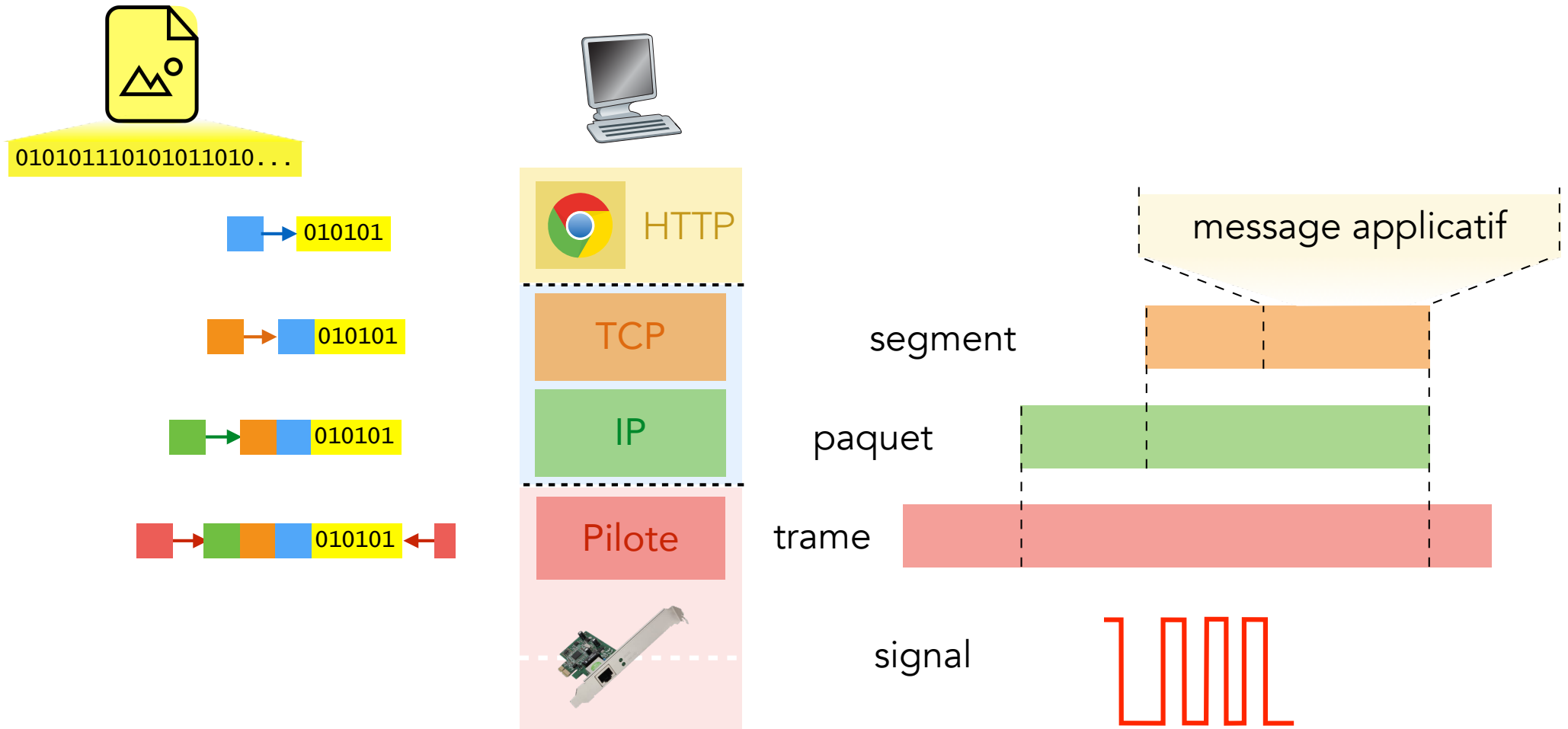
# Protocoles : les services rendus

## Contrôle de flux



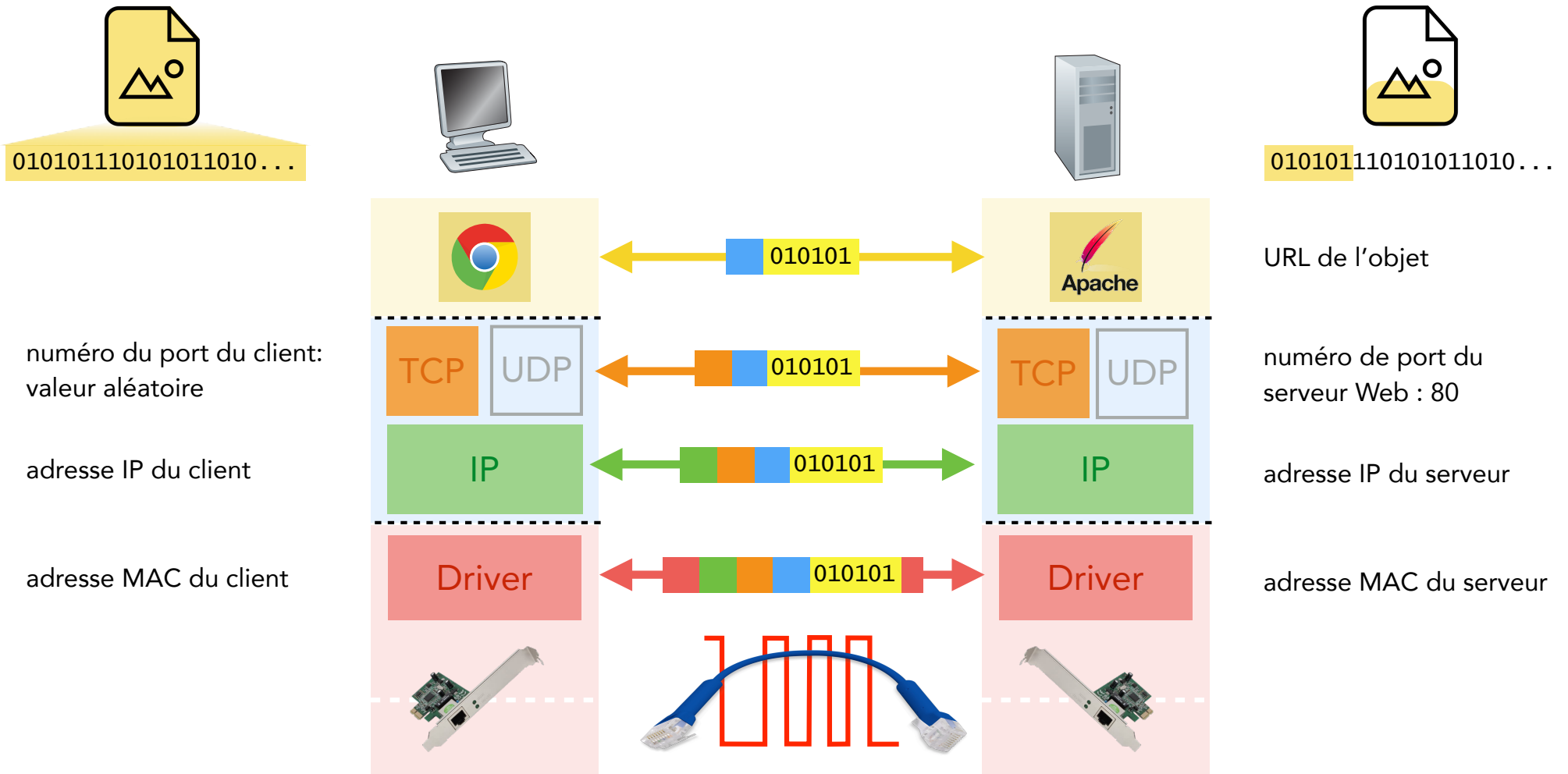
- Le récepteur annonce sa fenêtre de réception :
  - nombre de messages que la source peut transmettre avant de recevoir une nouvelle fenêtre
- La fenêtre évite les pertes qui adviendrait si le récepteur était engorgé

# Unité de données de protocole (PDU)

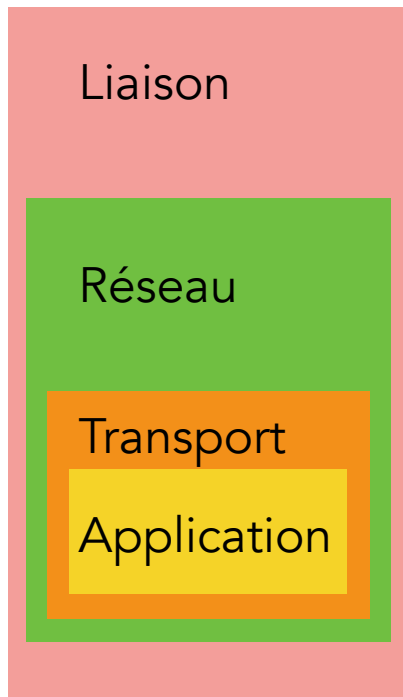


- Une PDU (protocol data unit) comprend l'entête du protocole, les données (et une enquee)
- Les données font référence à la PDU de la couche immédiatement supérieure (appelé SDU service data unit)
  - e.g. un paquet (couche 3) contient un segment (couche 4)

# Identifiants et adresses



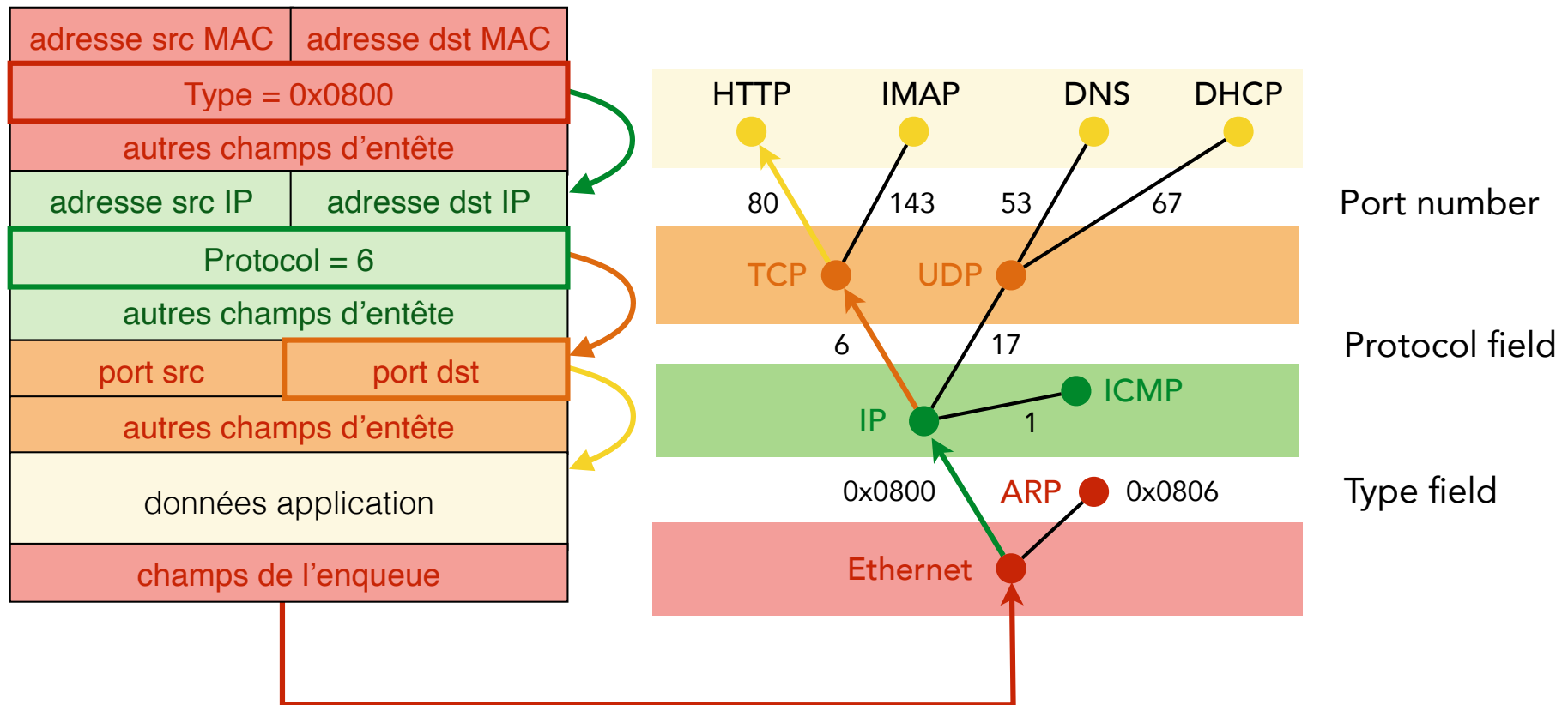
# Champs d'entête



adresse src MAC	adresse dst MAC	adresses MAC
Type = 0x0800		
autres champs d'entête		
adresse src IP	adresse dst IP	adresses IP
Protocol = 6		
autres champs d'entête		
port src	port dst	numéros de port
autres champs d'entête		
données application		noms (URL)
champs de l'enqueue		

- Une entête contient une suite de de champs :
  - des valeurs spécifiques dont la position indique la longueur et la signification
- Les données font référence à la PDU de la couche immédiatement supérieure (appelé SDU service data unit)
  - e.g. un paquet (couche 3) contient un segment (couche 4)

# Multiplexage/Démultiplexage

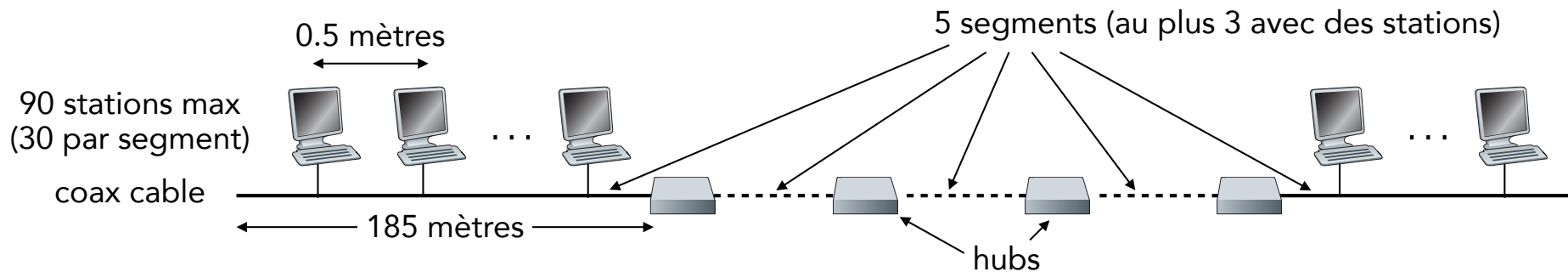


# Périphérie vs. Coeur de réseau

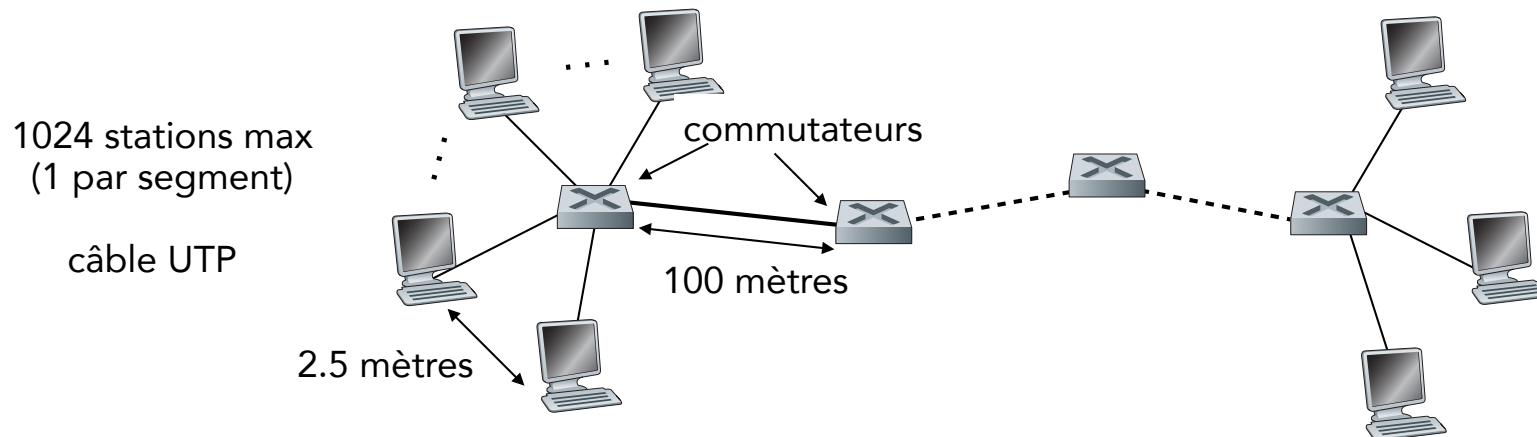
# LAN (Local Area Networks )

## Réseaux de petite/moyenne portée

### Bus

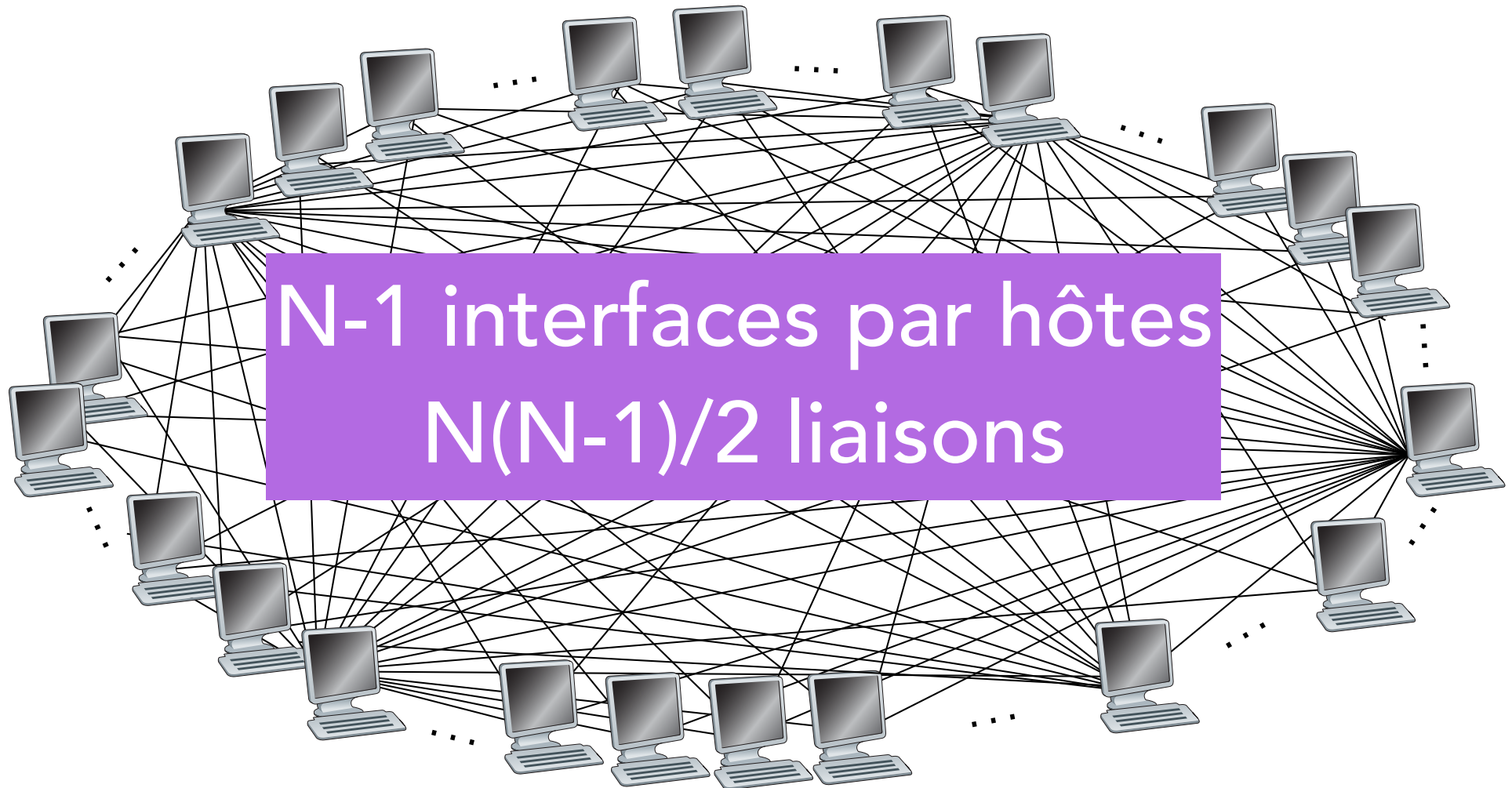


### Etoile



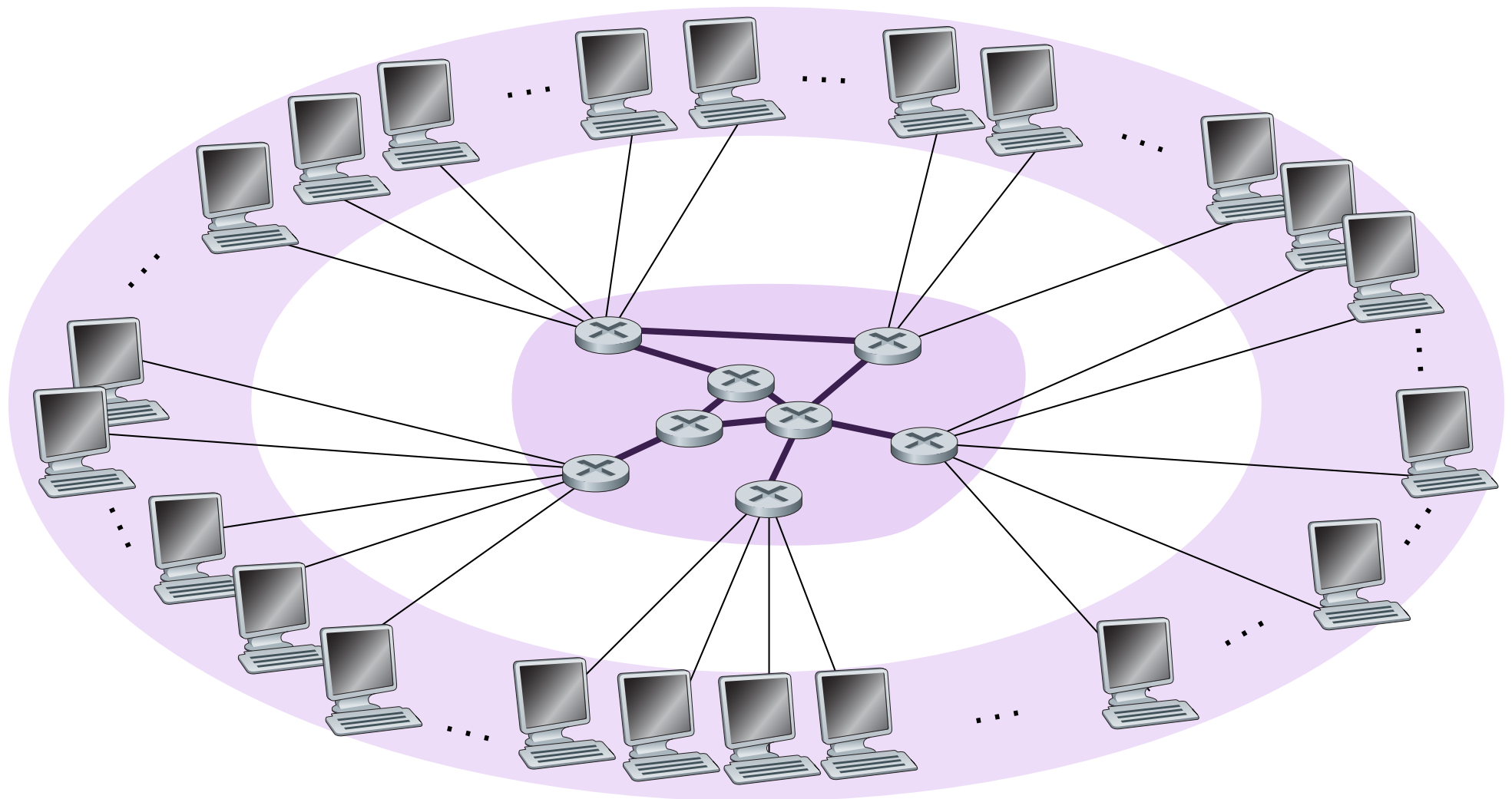


# WAN (Wide Area Networks) Réseaux de longue portée/distance



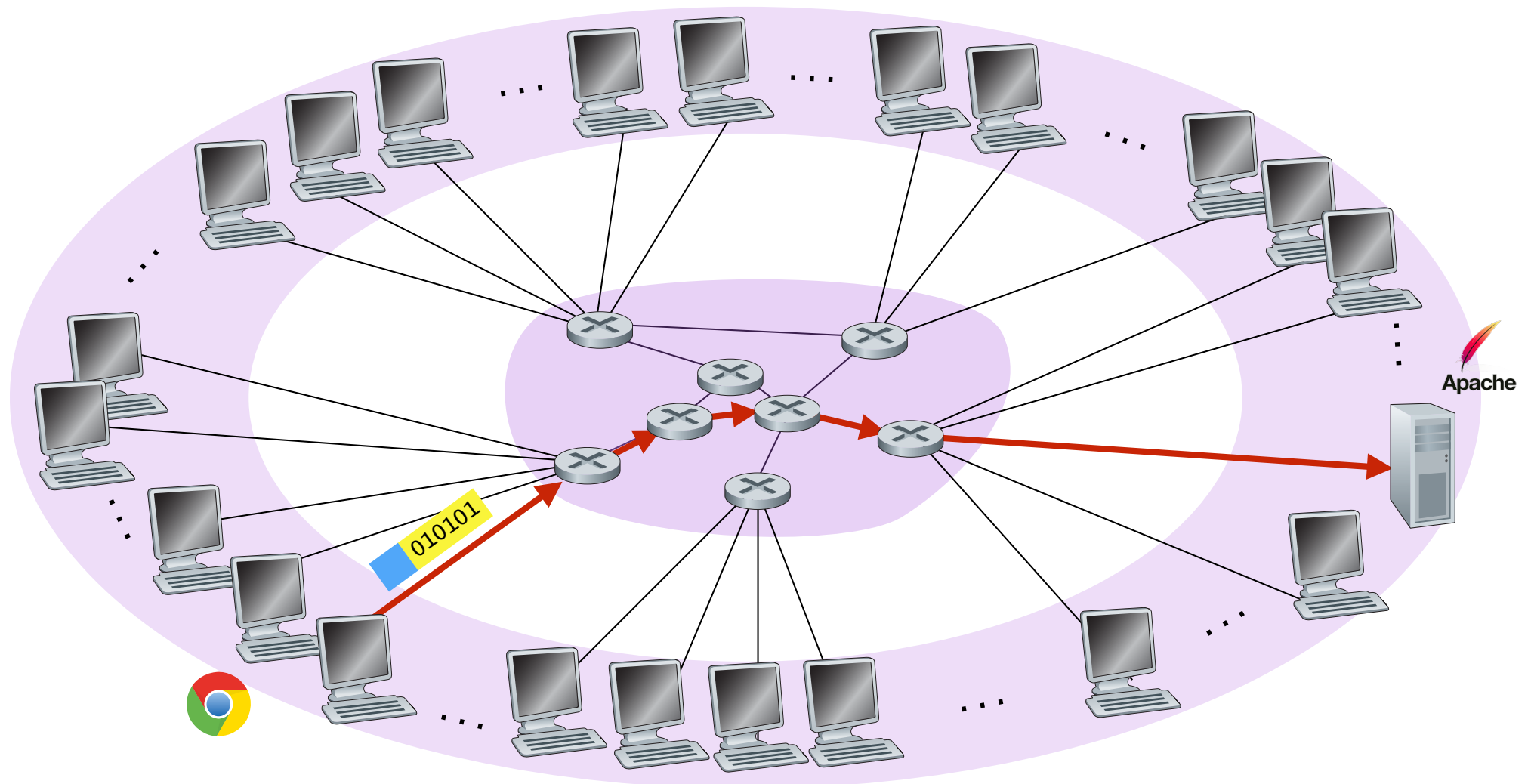
# Wide Area Networks

Périphérie versus Cœur



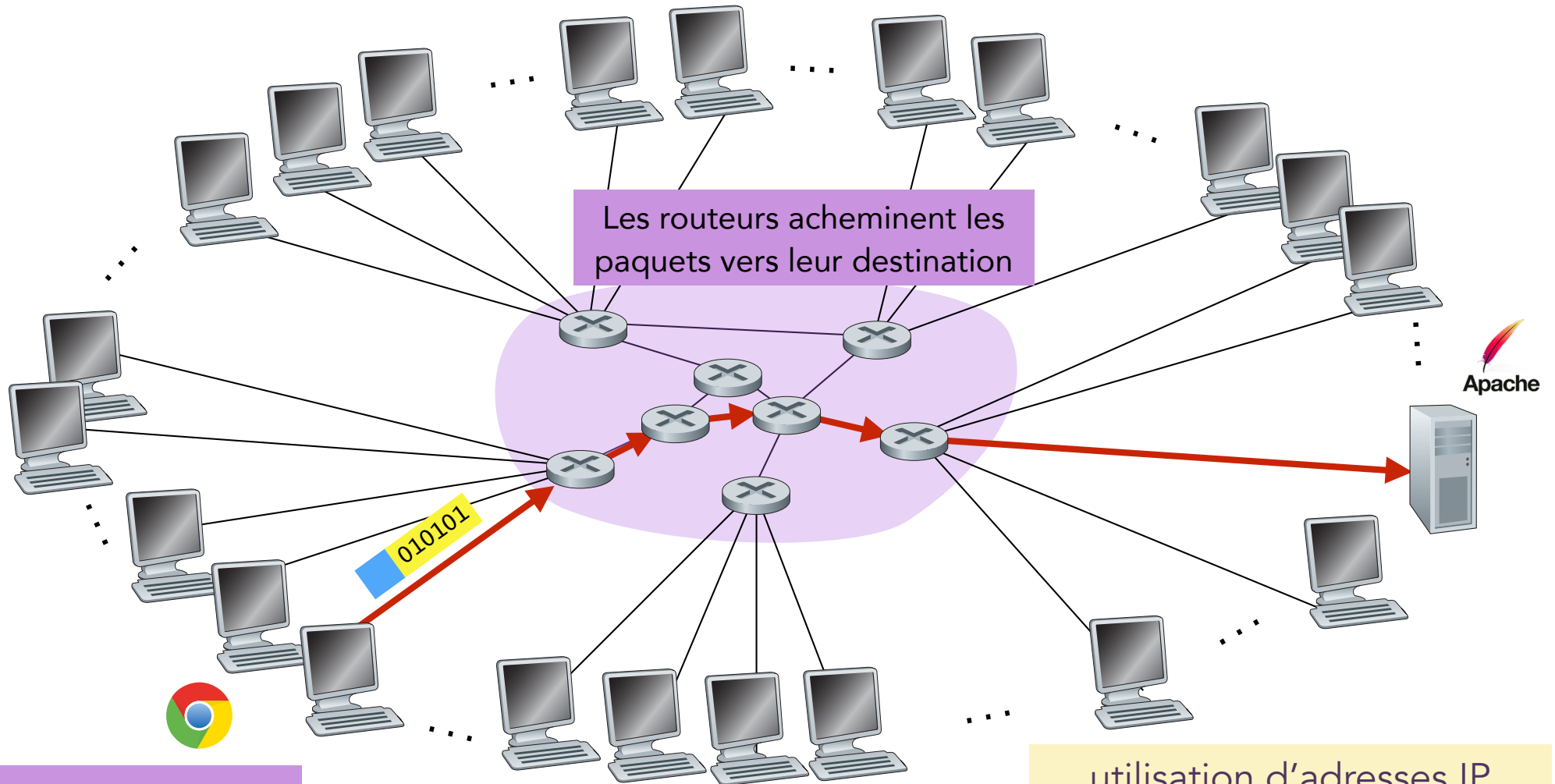
# Wide Area Networks

## Hôtes versus Routeurs



# Wide Area Networks

## Hôtes versus Routeurs



Les hôtes exécutent les applications

utilisation d'adresses IP

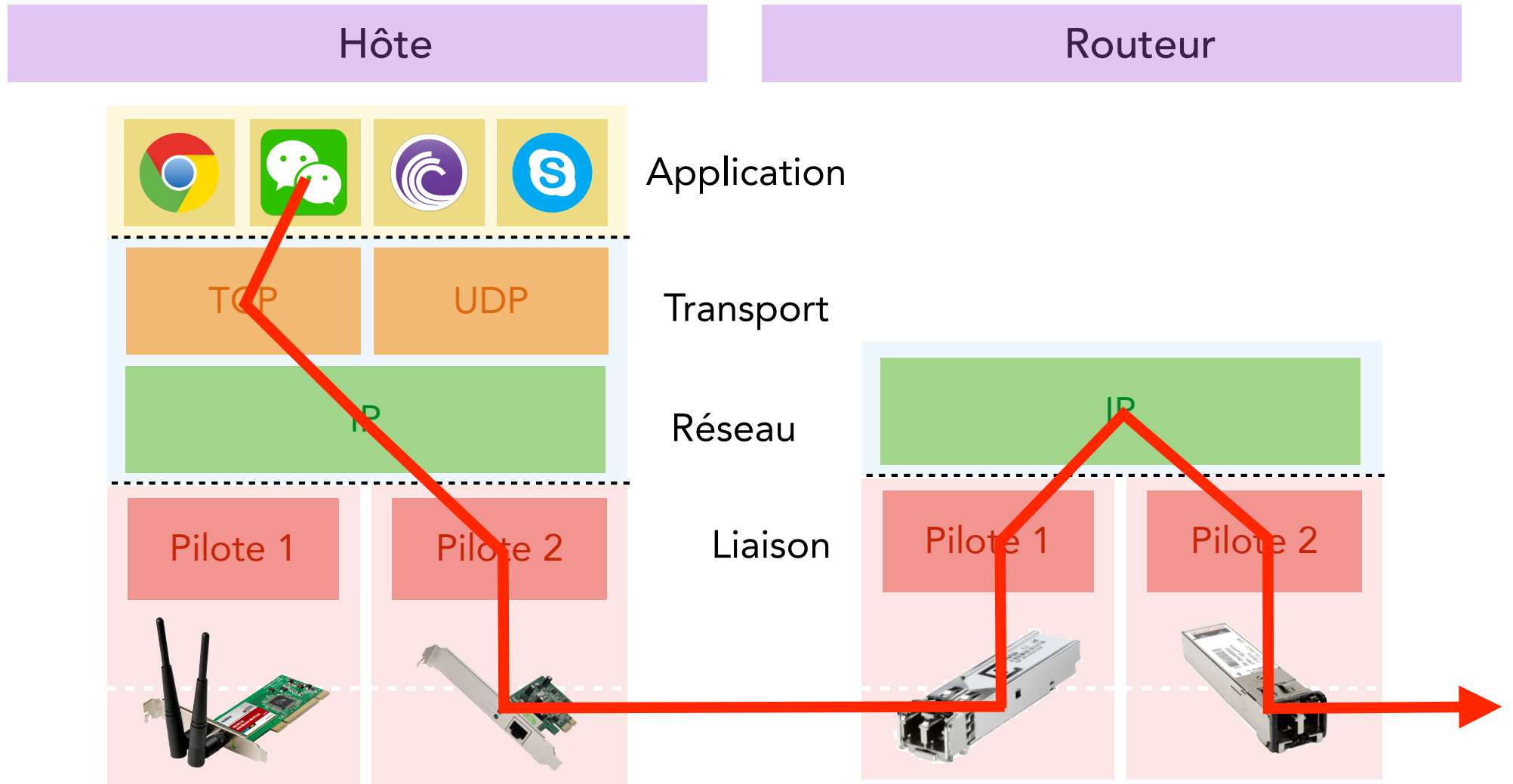
# Hôtes vs Routeur

## Répartition des tâches

- Routeur: Livraison du mieux possible
  - Examine l'adresse destination dans l'entête des paquets
  - Envoie le paquet dans la direction de la destination
- Hôte : Tout le reste
  - Exécute les applications réseaux
    - certaines applications veulent que leurs données soient reçues sans perte et dans l'ordre
  - Rattrape les dysfonctionnement du réseau :
    - retransmission des paquets perdus
    - remise en l'ordre des paquets
    - détection et réparation des paquets corrompus
    - évitement de la congestion du réseau et de l'engorgement du récepteur
- Ou non :
  - on utilise UDP

} TCP

# Pile protocolaire Internet



# Pile Protocolaire Internet

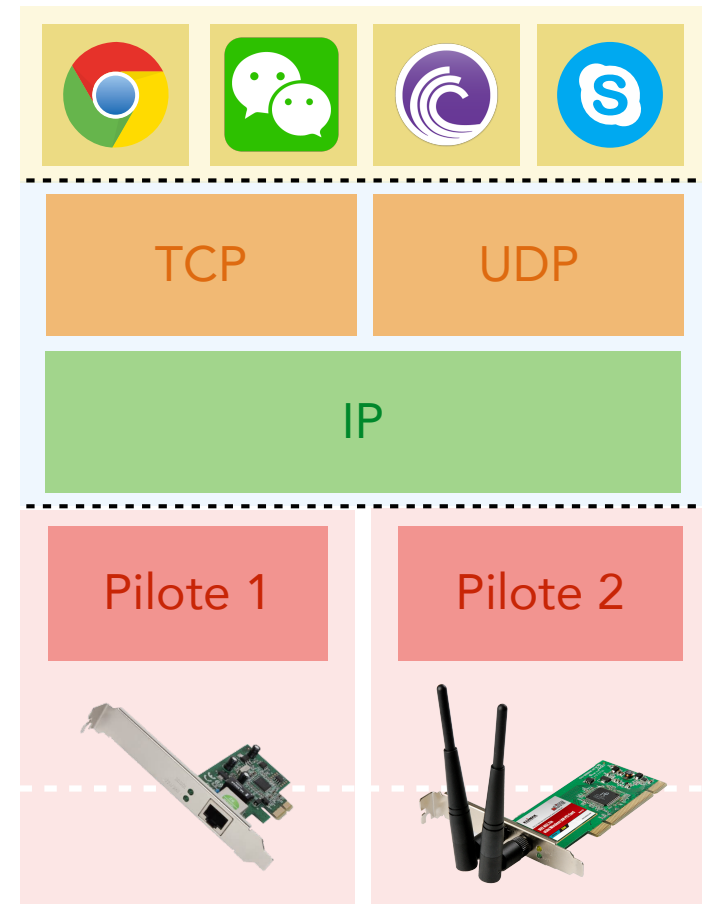
Application. Les utilisateurs peuvent obtenir des contenus ou des services

Transport. Transfert de données entre programmes applicatifs (fiable ou non)

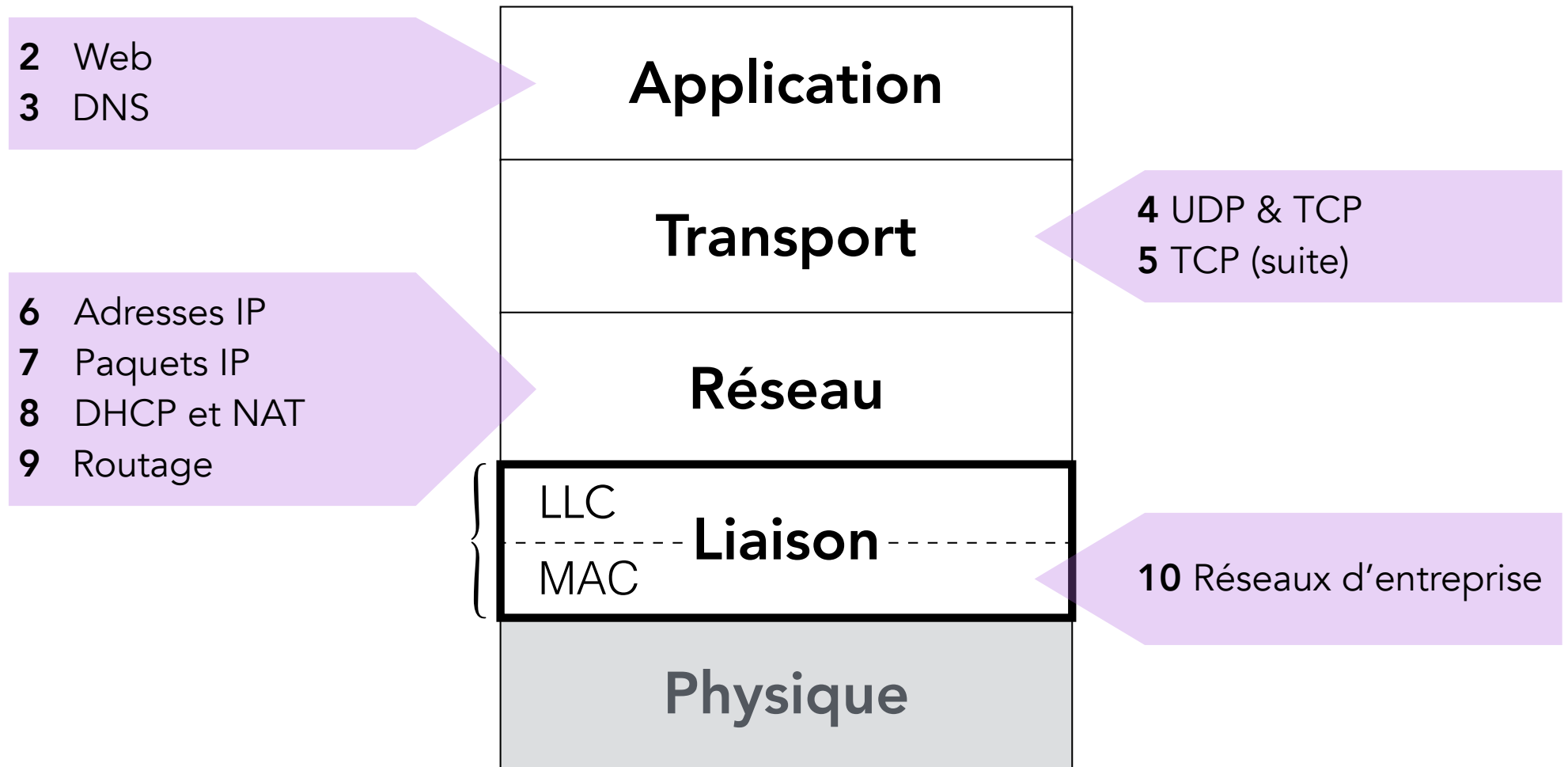
Réseau. Livraison de paquet depuis une machine source vers une machine destination

Liaison. Transfert de données entre stations voisines

Physique. Transmission de bits sous la forme d'un signal adapté au support de communication



# Chapitres suivants





# Conclusion

- Un réseau est conçu en définissant et distribuant les tâches à accomplir parmi :
  - les routeurs
  - les machines hôtes
- Les routeurs font :
  - du routage : sélection de route de l'acheminement des paquets de données
- Les machines hôtes exécutent
  - les applications
  - les fonctions de fiabilisation :
    - réparation et évitement de pertes, corrections d'erreurs
  - les fonctions d'utilisation efficace des ressources
    - calcul de la taille des messages, contrôle de congestion
- Cours prochain
  - HTTP, DNS