Architecture des réseaux - TD2

Michel Salomon

IUT de Belfort-Montbéliard
Département d'informatique

Protocoles mis en œuvre dans Internet

- Description
- Protocoles de la couche Transport
- Protocoles de la couche Réseau

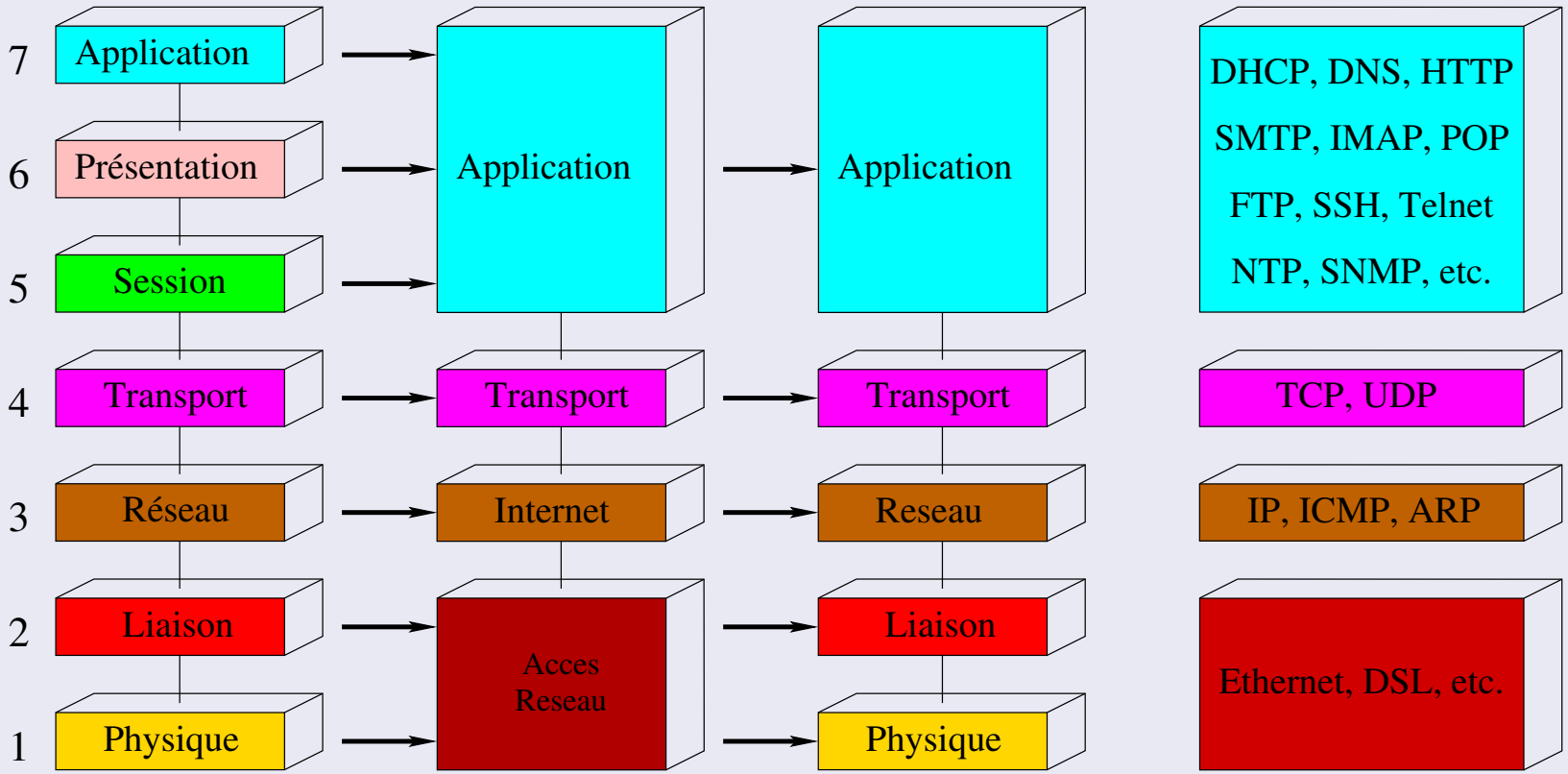
Description

- Suite de protocoles → l'ensemble des protocoles qui constituent la pile des protocoles utilisés dans Internet
- Souvent on parle aussi de pile ou modèle TCP/IP, du nom des deux protocoles majeurs → ils permettent de connecter deux hôtes appartenant à des réseaux différents
  - TCP → fait un transport fiable de bout en bout pour toute appli.
  - IP → se charge du routage à travers le réseau Internet

  Cependant, ce ne sont pas le seuls protocoles utilisés

Description

Du modèle OSI aux modèles Internet et TCP/IP



Couche Transport - *Transport Control Protocol*

Caractéristiques

- Transport de bout en bout en fragmentant si besoin les messages / données applicati(fs /ves) en segments
  - → Taille d'un segment dépendant de celle d'une trame (MTU)
- Mode connecté
  - → Établissement et rupture de connexion par poignées de mains
- Sans erreur
  - → Contrôle et retransmission si besoin
- Sans perte
  - → "Numérotation" et retransmission si besoin
- Avec séquencement
  - → Réassemblage des segments reçus dans le bon ordre grâce au numéro de séquence
- Avec contrôle de flux
  - → Le nœud source adapte son débit en émission au débit du nœud destination (prévient la congestion)

Couche Transport - *User Datagram Protocol*

Caractéristiques

- Transport sans fragmentation
- Mode non connecté
- Avec erreur et perte possibles
  - → Pas de contrôle, ni “numérotation” et aucune retransmission
- Sans séquencement
  - → Chaque segment UDP est indépendant des autres
- Sans contrôle de flux

Couche Transport - Identification des services offerts

Services identifiés par les numéros de port

- Un hôte (une machine), identifié(e) par son adresse IP, peut offrir des services applicatifs via l'un des deux protocoles
- Identification d'un processus serveur (ou démon)
  - Un numéro de port sur 16 bits
  - Un processus est un programme en cours d'exécution
- Une transmission (communication) entre deux processus
  (par ex. entre un client web - `firefox` et un serveur web - `apache2`)
  - En plus des adresses IP source et destination, il faut spécifier un port source et un port destination
- Interface système utilisée par un processus pour communiquer
  - Une **socket** (prise) donnant accès à la couche Réseau
  - Émission / réception (écrire / lire) sur la socket via le port
  - Identification d'un socket par `adresse_IP:numéro_de_port`

Couche Transport - Identification des services offerts

Services identifiés par les numéros de port (suite)

- Communication en mode connecté (TCP)
  - Connexion = "tuyau" entre 2 processus distants
  - Chaque processus a une socket à un bout du tuyau
  - Un couple de sockets définit la communication

    `193.52.61.144:3012` $\Leftrightarrow$ `193.52.61.138:80`

    Cet exemple définit une communication entre un client sur `193.52.61.144` ayant pris le port 3012 et le serveur à l'écoute sur le port 80, soit le serveur web, de l'hôte `193.52.61.138`
- Des services sont offerts via TCP, d'autres UDP, voire les 2
  - Un processus serveur (démon) écoute sur un port connu (cf. le fichier `/etc/services` pour la liste des ports connus)
  - Un processus client se voit attribué un port libre par le système

Couche Transport - Identification des services offerts

Affectation des ports

- Gérée par l'***I**nternet **A**ssigned **N**umbers **A**uthority*
- Numéro de port sur 16 bits → 65536 ports disponibles
- Numéros de port divisés en 3 groupes de ports
  1. Les ports connus compris entre 0 et 1023
     - 22 → `ssh`, 53 → `domain`, 80 → `http`, etc.
  2. Les ports en enregistrés entre 1024 et 49151
  3. Les ports privés entre 49152 et 65535

Couche Réseau - *Internet Protocol*

Caractéristiques

- Achemine des paquets (datagrammes) entre hôtes de réseaux diff.
  - Encapsulent des segments
  - Durée de vie limitée (champ ***T**ime **T**o **L**ive* dans l'en-tête IP)
- De manière indépendante, sans connexion, via différents types de réseaux (Couche Liaison de données → Ethernet, FDDI, ATM, etc.)
- Sans garantir l'arrivée à destination
- Sans imposer de délai de transmission (***Q**uality **o**f **S**ervice*)
- Un hôte / nœud est identifié par une adresse IP
  - Adresses IP publiques attribuées par l'IANA
  - Met en œuvre les décisions de l'***I**nternet **C**orporation for **A**ssigned **N**ames and **N**umbers*
- Acheminement distant assuré par les routeurs
  - Utilisent des tables de routage
  - N'ont qu'une connaissance locale du réseau (routeurs voisins)
  - Font au "mieux" (politique *best effort*)

Couche Réseau - *Internet Protocol*

Adresse IP (32 bits en IPv4)

- Définit implicitement
  - **Une adresse IP de réseau** (`network`)
  - **un numéro (ou adresse) d'hôte** dans ce réseau

  via un **préfixe** (IPv4 / 6) ou **masque de réseaux** (*netmask* - IPv4)

- Tout réseau a une adresse IP qui l'identifie
  $\rightarrow$ termine par un ou plusieurs bits à 0

- Existence d'adresses IP désignant simultanément plusieurs hôtes $\rightarrow$ adresse de diffusion ou *broadcast* $\rightarrow$ 2 types de com.
  1. Point à point ou *unicast* $\rightarrow$ 1 source et 1 destination
  2. Point à multipoint ou *broadcast* $\rightarrow$ 1 source et plusieurs dest.

Cette approche permet a un hôte source de déterminer si l'hôte destination est dans même réseau que lui ou non (envoi au routeur)

Couche Réseau - *Internet Protocol*

Existence d'adresses IP dites privées ou réservées

- Non routées sur Internet → paquets détruits par certains routeurs
- Notation utilisée dite *décimale pointée*
  - `0.0.0.0` → désigne la route par défaut
  - `127.X.X.X` → boucle locale
    Permet à une machine de s'"envoyer" des paquets, même sans carte / interface réseau, `127.0.0.1` = adresse de *loopback*
  - `10.0.0.1` à `10.255.255.254` (préfixe = 8)
    → vastes réseaux privés
  - `172.16.0.1` à `172.31.255.254` (préfixe = 12)
    → moyens réseaux privés
  - `192.168.0.1` à `192.168.255.254` (préfixe = 16)
    → petits réseaux privés

  Adresses utilisées pour créer des réseaux privés accédant à Internet via la translation d'adresse (***N**etwork **A**ddress **T**ranslation*)
- Blocs d'adresses anciennement associés aux classes de réseaux A,B et C → obsolète (aujourd'hui ***C**lassless **I**nter**D**omain **R**outing*)

Couche Réseau - *Internet Control Message Protocol*

Caractéristiques

- Permet de faire du contrôle d'erreur et de la gestion d'erreurs lors de l'acheminement d'un datagramme via IP
- Par exemple lorsqu'un hôte n'est pas accessible
- Différents types de message ICMP (un message est encapsulé dans un datagramme IP)
- Permet à un hôte / un routeur d'obtenir des informations ou d'indiquer une erreur (mise à jour des tables de routage)

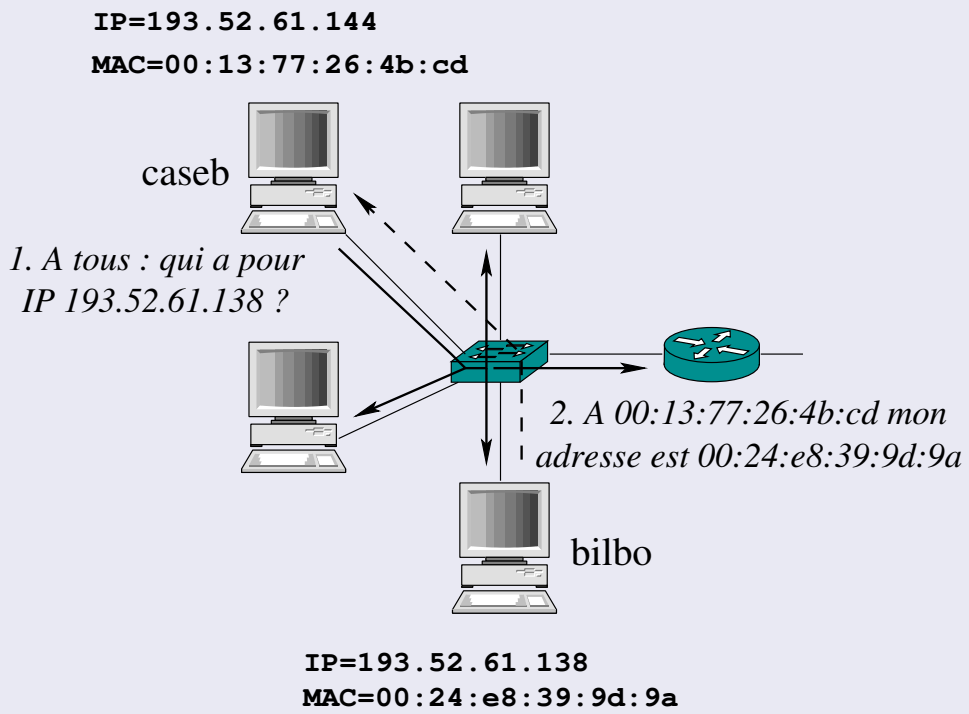
Quelques commandes l'utilisant

- `ping` (permet d'obtenir le ***R**ound **T**rip **T**ime*)
  - Message `ICMP_ECHO_REQUEST` de la source $\rightarrow$ la destination
  - Message `ICMP_ECHO_REPLY` de la destination $\rightarrow$ la source
- `traceroute`, `netstat`

Couche Réseau - *Address Resolution Protocol*

Caractéristiques

- Permet de trouver une adresse MAC / physique (par ex. Ethernet) à partir d'une adresse IP / logique
- Stockage des correspondance dans une table locale sur l'hôte dans le cache ARP (peut être affichée avec la commande `arp`)
- Obtention de l'adresse MAC par *broadcast*, puis *unicast*



Exercices sur les adresses et la configuration

Exercice 1 - Translation d'adresses

- Donner les différents types d'adresses permettant d'identifier une machine et pour chacune la couche où elle est utilisée
- Indiquer les protocoles qui permettent de passer d'un type d'adresse à un autre, du moins quand ils existent

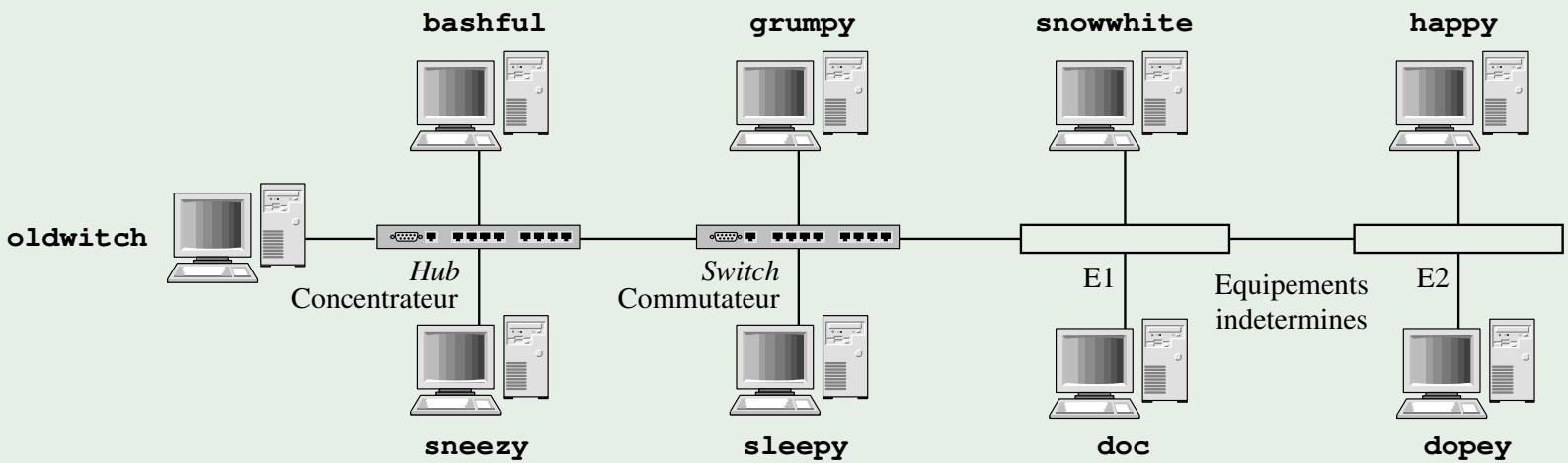
Exercice 2 - Configuration réseau d'une machine

- Quelles informations sont nécessaires pour configurer l'accès réseau d'une machine ? *Identification de la machine ; routage ; etc.*
- Quel est le rôle du préfixe / masque de réseau ?
- Sous Linux, quels fichiers contiennent ces informations ?

Exercices sur les adresses et la configuration

Exercice 3 - Communications et équipements d'interconnexion

- On considère l'architecture de réseau ci-dessous



formé de 9 machines reliées par 4 équipements d'interconnex. dont 2 sont indéterminés (*E1* et *E2*)

- Bien que l'on parle que d'un réseau, il peut y avoir plusieurs sous-réseaux au sein de celui-ci

Exercices sur les adresses et la configuration

Exercice 3 (suite)

- On sait d'autre part que la masque utilisé dans ce réseau est `255.255.255.248` et que les adresses IP

  → `192.168.0.1`,`192.168.0.6`, `192.168.0.12` et `192.168.0.18`

  sont attribuées aux machines

  → `doc`, `happy`, `grumpy` et `sleepy`

  mais sans savoir quelle adresse est à quelle machine

- On vous demande de répondre aux questions suivantes :
  1. Lorsqu'un paquet de diffusion ARP est émis par `sneezy`, quelles machines recevront ce paquet ?
  2. Lorsqu'un paquet est émis par `sneezy` vers la machine `bashful`, quelles machines recevront ce paquet ?
  3. Lorsqu'un paquet est émis par `sneezy` vers la machine `sleepy`, quelles machines recevront ce paquet ?
  4. Quelle est la valeur du préfixe ?

Exercices sur les adresses et la configuration

Exercice 3 (suite)

- On sait d'autre part que la masque utilisé dans ce réseau est `255.255.255.248` et que les adresses IP

  → `192.168.0.1`,`192.168.0.6`, `192.168.0.12` et `192.168.0.18`

  sont attribuées aux machines

  → `doc`, `happy`, `grumpy` et `sleepy`

  mais sans savoir quelle adresse est à quelle machine

- On vous demande de répondre aux questions suivantes :
  5. À partir des adresses IP et du masque de réseau proposer une façon d'assigner chacune des adresses IP à l'une des machines, puis en déduire la nature de l'équipement d'interconnexion *E1*
  6. Enfin, sachant que l'adresse de diffusion du réseau de la machine `dopey` est `192.168.0.23` en déduire une affectation possible pour les 4 adresses IP et la nature de l'équipement d'interconnexion *E2*