

Troisième cours : Réseaux, adresses, protocoles

Réseaux et adresses
 Le concept de réseau
 Adresses et protocoles de réseau

Protocoles et moyens de communication
 Protocoles simples
 Protocoles plus élaborés

Courrier électronique
 Principes du courrier électronique
 Le protocole SMTP
 Forme d'un message

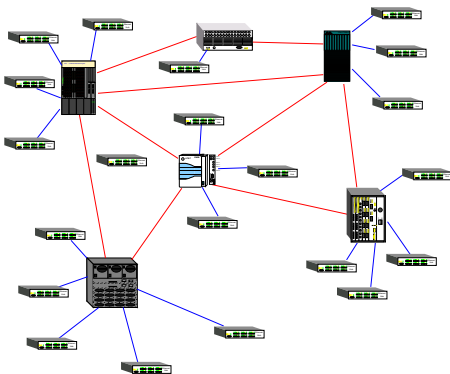
Systèmes informatiques

Olivier Lecarme

Licence Mathématiques-Informatique, Semestre 2

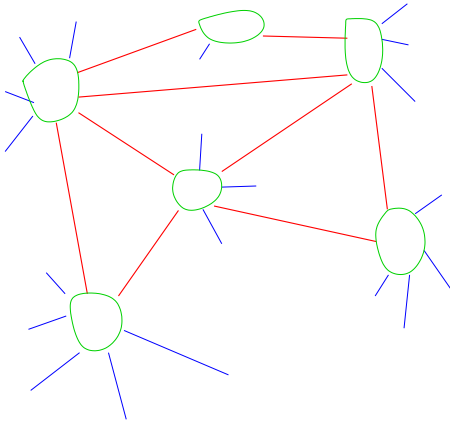
2006-2007

Le concept de réseau



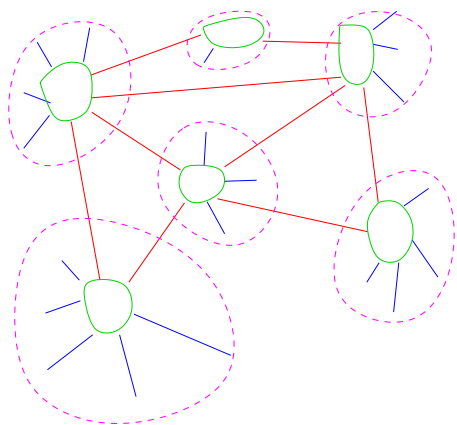
- ▶ un réseau est un ensemble de nœuds (sommets) et de liens (arêtes)
- ▶ on peut donc le modéliser par un graphe

Le concept de réseau



- ▶ analogie possible avec un réseau routier
- ▶ routes, échangeurs, convois

Le concept de réseau



- ▶ les sous-réseaux peuvent correspondre à des opérateurs distincts
- ▶ ils sont nationaux ou trans-nationaux
- ▶ ils sont reliés par leurs nœuds centraux



Le concept de réseau (suite)

- ▶ un *réseau* relie entre eux :
 - ▶ des ordinateurs
 - ▶ des postes de travail
 - ▶ des imprimantes ou autres périphériques partagés
 - ▶ des nœuds de communication
 - ▶ etc.
- ▶ le terme est très générique :
 - ▶ **mécanismes** de communication
 - ▶ **techniques** de communication
 - ▶ **ensemble** des composants



Types de réseaux

- ▶ *réseau local* :
 - ▶ relie des ordinateurs d'un particulier, d'un bâtiment, d'une entreprise
 - ▶ utilise des lignes téléphoniques, des liaisons radio, des fibres optiques, des câbles coaxiaux
 - ▶ exemple : le **réseau du parc Valrose**
- ▶ *réseau étendu* :
 - ▶ relie des ordinateurs **distants** de centaines ou milliers de kilomètres
 - ▶ utilise des câbles, fibres optiques, lignes spécialisées, liaisons par satellite, les réseaux publics de télécommunication
 - ▶ exemples : le **réseau de l'UNSA**, relié au réseau R3T2 de la région PACA, relié au réseau national RENATER, relié aux autres réseaux nationaux

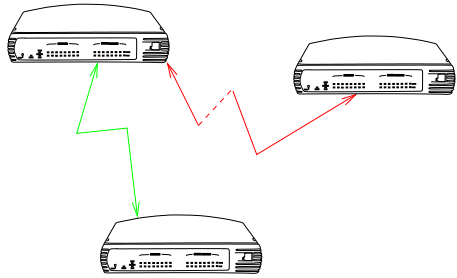


Transmission des informations

- ▶ plusieurs manières possibles :
 - ▶ *liaison directe*, comme dans un interphone
 - ▶ *diffusion*, comme dans un réseau de radiophonie
 - ▶ *transmission*, comme dans une conversation téléphonique



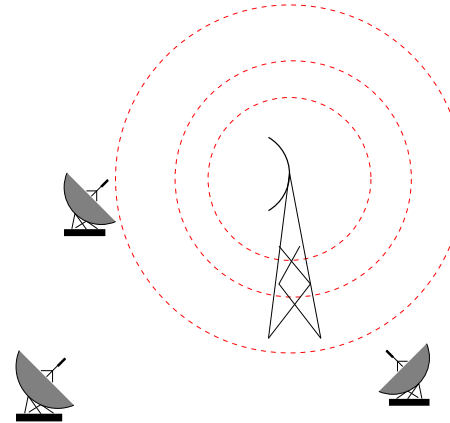
Liaison directe



- ▶ on **établit** la connexion entre **deux** postes
- ▶ elle **empêche** toute autre connexion
- ▶ possible uniquement dans un **réseau très simple**
- ▶ utilisée surtout pour une **liaison fixe**



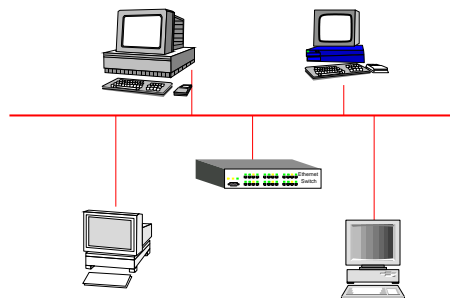
Diffusion



- ▶ l'émetteur envoie dans **toutes les directions**
- ▶ tous les **récepteurs** captent l'information
- ▶ seuls ses **destinataires** l'utilisent



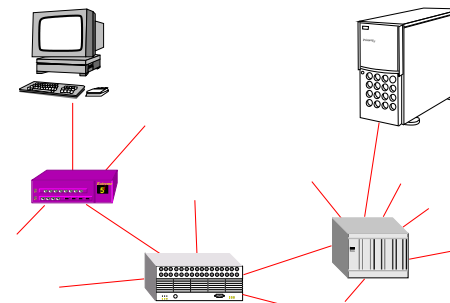
Réseau Ethernet



- ▶ mise en application du **principe de diffusion**
- ▶ tout organe a une **adresse Ethernet** caractéristique dans le monde entier
- ▶ chacun est **à l'écoute** de ce qui le concerne



Transmission



- ▶ la liaison passe par plusieurs **commutateurs**
- ▶ l'établissement de la liaison est **temporaire**



Transmission de paquets

- ▶ chaque commutateur a un **nombre limité** de canaux disponibles
- ▶ on ne peut pas **bloquer tous ces canaux** pour une transmission de durée imprévisible
- ▶ on utilise donc la **transmission de paquets** :
 - ▶ l'information (le *message*) est découpée en fragments (les *paquets*)
 - ▶ les paquets peuvent être transmis **indépendamment** les uns des autres
 - ▶ il est donc nécessaire de les **regrouper** à l'arrivée pour **reconstituer** le message
- ▶ analogie avec des convois routiers ou des trains



Internet

- ▶ **réseau des réseaux**
- ▶ relie entre eux les **réseaux nationaux** de tous les pays, mais aussi les réseaux non nationaux des opérateurs régionaux, à longue distance ou publics
- ▶ des **protocoles nombreux** règlent les échanges d'informations :
 - ▶ fichiers
 - ▶ groupes de discussion
 - ▶ courrier électronique
 - ▶ connexion à distance
 - ▶ etc.



Protocole

- ▶ un **protocole** intervient dans tout **échange de données** entre ordinateurs :
 - ▶ forme des données
 - ▶ codification utilisée
 - ▶ conventions pour demander la répétition
 - ▶ conventions pour l'accusé de réception
 - ▶ etc.
- ▶ les deux protocoles **les plus utilisés** sont **TCP** (*Transmission Control Protocol*) et **IP** (*Internet Protocol*), développés ensemble vers la fin des années 1970



Internet (suite)

- ▶ **démarrage** en 1983
- ▶ évolution très rapide :
 - ▶ moins de 100 000 postes en 1989
 - ▶ 5 millions en 1995
 - ▶ 40 millions en 1998
 - ▶ 400 millions au début de 2006
- ▶ les **principes de base** n'ont guère changé
 - ▶ certains protocoles s'avèrent vraiment **inadaptés** (courrier)
 - ▶ d'autres protocoles sont utilisés de manière **imprévue** (HTTP)



Routeurs et adresses

- ▶ les réseaux communiquent entre eux par l'intermédiaire de **nœuds d'interconnexion (routeurs)**
- ▶ un message passe par **plusieurs routeurs** avant d'arriver au réseau local de l'ordinateur destinataire
- ▶ les messages sont **découpés en paquets**, transmis par des chemins qui peuvent **varier d'un paquet à l'autre**
- ▶ sur le réseau local l'ordinateur a une **adresse physique**
- ▶ le message comporte une **adresse IP**
- ▶ le réseau local fait se **correspondre les deux adresses**
- ▶ un routeur envoie un message vers le **routeur voisin** capable de traiter l'adresse IP



Adresse IP (suite)

- ▶ **classe C** :
 - ▶ $192 \leq \text{octet } 1 < 224$
 - ▶ octets 1 à 3 caractéristiques du réseau
 - ▶ octet 4 identifiant les machines
 - ▶ plus de 2 millions de petits réseaux, 256 machines par réseau
- ▶ beaucoup d'entreprises **cachent** les adresses IP internes
- ▶ malgré tout le système actuel **atteint ses limites**
- ▶ le système **IPv6** doit le remplacer à terme



Adresse IP

- ▶ **nombre de 32 bits** (actuellement), découpé en **quatre octets**
- ▶ les réseaux membres d'Internet sont de **trois classes** :
 - ▶ **classe A** :
 - ▶ octet 1 < 128
 - ▶ les trois octets suivants identifient les machines
 - ▶ pas plus de 128 très grands réseaux
 - ▶ **18** représente le réseau du MIT
 - ▶ **classe B** :
 - ▶ $128 \leq \text{octet } 1 < 192$
 - ▶ octets 1 et 2 caractéristiques du réseau
 - ▶ les deux derniers octets identifient les machines
 - ▶ 16384 réseaux, 65536 machines par réseau
 - ▶ **134.59** représente le réseau de l'UNSA



Domaines et adresses symboliques

- ▶ l'adresse IP est **malcommode** pour l'utilisateur
- ▶ on définit des **adresses symboliques** hiérarchisées en **domaines**
- ▶ des **serveurs de noms** (ordinateurs ou services spécialisés) établissent la **correspondance** depuis et vers les adresses IP
- ▶ système **hiérarchisé** :
 - ▶ le **niveau le plus élevé** apparaît en dernier :
 - ▶ un domaine par pays : **fr**, **uk**, **za**, etc.
 - ▶ domaines « apatrides » : **org**, **net**, **com**, **biz**, etc.
 - ▶ domaines réservés aux USA : **mil**, **gov**, **edu**
 - ▶ les niveaux précédents sont gérés par l'organisme propriétaire du domaine
- ▶ il existe souvent des **alias** pour différents noms



Exemples d'adresses symboliques

- ▶ **unice.fr** est le domaine de l'UNSA :
 - ▶ toutes les machines de Valrose sont accessibles sous la forme *nom de machine.unice.fr*
 - ▶ **sesame-mips.unice.fr** est un alias pour **mistral.unice.fr**, de numéro IP **134.59.2.10**
 - ▶ **www-mips.unice.fr** est un alias pour **corbieres.unice.fr**, de numéro IP **134.59.2.154**
 - ▶ les machines d'une partie du laboratoire I3S sont dans un **sous-domaine** de numéro **134.59.133**
 - ▶ la machine **zenon.i3s.unice.fr** a pour adresse IP **134.59.133.2**
- ▶ le domaine **mit.edu** est celui du *Massachusetts Institute of Technology*, aux USA
- ▶ le nom **mit.edu** est aussi un alias pour une machine d'adresse IP **18.7.22.69**, nommée **web.mit.edu**

Le protocole TCP

- ▶ il intervient aux **deux extrémités** de la communication :
 - ▶ **découpage** du message en paquets au départ
 - ▶ association à chaque paquet d'une *enveloppe* identifiant l'expéditeur, le destinataire, le message lui-même et l'ordre des paquets
 - ▶ **réassemblage** des paquets à l'arrivée
- ▶ il assure la **sécurité** et l'**intégrité** de la transmission
- ▶ dialogue avec l'expéditeur pour redemander les **paquets manquants ou mal transmis**
- ▶ sert de support à de très nombreuses **applications de communication**
- ▶ chaque application utilise un **numéro de port spécialisé** qui l'identifie
- ▶ ainsi le message à l'arrivée est **transmis à la bonne application**

Quelques outils simples

- ▶ association entre **nom symbolique** et **adresse IP** :
 - ▶ il s'agit donc d'interroger un *serveur de noms* (normalement choisi par défaut)
 - ▶ la commande **host** a pour argument **une des deux adresses** et répond avec l'autre
 - ▶ la commande **dig** fournit plus d'informations et a plus de possibilités
- ▶ test d'**accessibilité** d'une machine
 - ▶ la commande **ping** a pour argument la **machine à tester**
 - ▶ on peut l'utiliser pour **tester des relais** de plus en plus éloignés et déterminer où apparaît une panne
 - ▶ beaucoup de machines **n'y répondent pas** pour raisons de sécurité
- ▶ la commande **traceroute** montre le **chemin de routeur en routeur** vers une machine donnée

Le concept de protocole

- ▶ l'**appel téléphonique** respecte un protocole :
 - ▶ l'appelé indique qu'il a **reçu l'appel** : « Allo? »
 - ▶ l'appelant vérifie qu'il est en **communication avec la bonne personne** : « C'est bien la boucherie Sanzot? »
 - ▶ l'appelé **s'identifie** : « Ici le capitaine Haddock »
 - ▶ l'appelant **s'identifie** : « Ici Séraphin Lampion »
 - ▶ et le dialogue peut commencer
- ▶ un protocole final sert à **terminer la communication**
- ▶ en cas de très mauvaise transmission, il faut un protocole très élaboré pour que le dialogue puisse se faire (dialogue de sourds)

Exemple élémentaire

- ▶ la commande `write` permet d'envoyer du **texte sur le terminal de quelqu'un d'autre**
- ▶ **protocole à suivre** entre Toto et Jojo :
 - ▶ Toto obtient le **nom d'un terminal** sur lequel Jojo est connecté : commande `who jojo`
 - ▶ Toto tape `write jojo nom de terminal`
 - ▶ Jojo voit le **message annonçant un dialogue** venant de Toto
 - ▶ Jojo répond par la commande `write toto nom de terminal`
 - ▶ à partir de ce moment, **tout ce que tape l'un** apparaît dans le **Xterm de l'autre**
 - ▶ pour éviter le mélange des messages, chacun doit **indiquer que son message est terminé**, puis attendre son tour
 - ▶ chacun **termine le dialogue** par l'indication de fin de fichier (`C-d`)



Inconvénients des protocoles trop simples

- ▶ les deux interlocuteurs doivent être **présents simultanément**
- ▶ avec `write` ils doivent être sur la **même machine**
- ▶ avec `talk` il faut la compatibilité entre machines
- ▶ un interlocuteur peut **refuser la communication** :
 - ▶ la commande `mesg n` refuse les dialogues dans ce Xterm
 - ▶ `mesg y` les accepte
 - ▶ `mesg` indique l'état actuel
- ▶ ces deux protocoles servent à la **communication d'une personne avec une autre**, sous forme **immédiate**



Exemple plus perfectionné

- ▶ la commande `talk` gère elle-même son protocole :
 - ▶ les fenêtres Xterm sont **découpées en deux parties**
 - ▶ celle du haut contient le texte tapé par l'**utilisateur**
 - ▶ celle du bas contient le texte tapé par l'**interlocuteur**
 - ▶ tous deux peuvent **taper simultanément**
 - ▶ les deux interlocuteurs peuvent être sur des **machines différentes**
- ▶ malheureusement :
 - ▶ beaucoup de machines **refusent le protocole talk**
 - ▶ dans beaucoup de cas la communication est **impossible entre deux machines distantes**



Le protocole IRC

- ▶ IRC = *Internet Relay Channel*
- ▶ permet la **communication entre plusieurs personnes**, sous forme **immédiate**
- ▶ **architecture** élaborée :
 - ▶ des **serveurs** IRC sont reliés entre eux
 - ▶ un utilisateur se connecte à un serveur
 - ▶ il choisit un **canal** parmi un très grand nombre
 - ▶ **liberté** de définir de nouveaux canaux
 - ▶ tout message tapé sur un canal est transmis par le serveur à **tous les autres serveurs** qui servent ce canal
 - ▶ tout le monde **parle ensemble**, comme dans une foule
 - ▶ **peu de gens écoutent**
- ▶ plusieurs **outils** plus ou moins perfectionnés
- ▶ les navigateurs sont utilisés pour gérer des **protocoles similaires (chat)**



Le protocole NNTP

- ▶ NNTP = *Network News Transfer Protocol*
- ▶ permet la **communication entre plusieurs personnes**, sous forme **différée**
- ▶ apparenté à des **panneaux d'affichage publics**
- ▶ **architecture** également élaborée :
 - ▶ des **serveurs NNTP** communiquent entre eux
 - ▶ un utilisateur peut s'**abonner** à un ou plusieurs **groupes de discussion**
 - ▶ il peut **lire tous les articles** récents publiés dans ces groupes
 - ▶ il peut **envoyer des articles** destinés à ces groupes
 - ▶ les serveurs se **synchronisent** régulièrement pour transmettre les nouveaux articles
 - ▶ les **administrateurs des serveurs** choisissent les groupes qu'ils diffusent
 - ▶ ils choisissent aussi la **durée de vie des articles**

Fonctionnement du protocole NNTP

- ▶ il existe des **outils spécialisés** dits **lecteurs de nouvelles**
- ▶ on peut souvent aussi passer par un **navigateur**
- ▶ on dépend très fortement du **serveur NNTP** auquel on a accès (un seul en général)
- ▶ les lecteurs sont des **clients** du serveur
- ▶ le **ystème Usenet** est l'ensemble des serveurs NNTP
- ▶ remarques :
 - ▶ le nombre de groupes est **gigantesque**
 - ▶ le serveur auquel on est connecté ne les sert **certainement pas tous**
 - ▶ on ne peut pas s'abonner raisonnablement à **plus d'une dizaine**
 - ▶ le rapport signal sur bruit est **très proche de zéro**

Particularités du protocole NNTP

- ▶ les articles sont **publics**
- ▶ les participants n'ont pas besoin d'être connectés **simultanément**
- ▶ les articles sont **conservés** un certain temps (variable)
- ▶ il n'y a pas d'**identification absolue** des articles
- ▶ certains groupes ont un **modérateur** qui **filtre** les articles
- ▶ le nombre de groupes existants est gigantesque : **plusieurs dizaines de milliers!**
- ▶ les groupes sont organisés suivant une **hiérarchie** :
 - ▶ terme le plus général en **premier** : **comp, rec, alt, fr, unice**, etc.
 - ▶ un ou plusieurs **termes descriptifs** : **unice.ragots**

Protocoles de messagerie instantanée

- ▶ mécanisme de **communication immédiate entre personnes**
- ▶ nécessité de **présence simultanée** des deux interlocuteurs
- ▶ similaire aux conversations privées du protocole IRC
- ▶ nombreux **protocoles privés** des divers fournisseurs de services (*AIM, ICQ, MSN, Yahoo*)
- ▶ protocole *Jabber* du monde des logiciels libres
- ▶ les protocoles privés sont **incompatibles entre eux**
- ▶ les outils de logiciel libre sont **compatibles avec tous les protocoles**

Outils de messagerie instantanée

- ▶ dans le monde de Windows, chacun utilise l'outil de son fournisseur de services, ce qui sert de **moyen de fidélisation**
- ▶ il existe des **outils multi-protocoles** dans le monde du logiciel libre
- ▶ *Gaim* est le plus général, utilisable sur **la plupart des plate-formes** et avec **la plupart des protocoles**
- ▶ *Gabber* est l'outil de Unix pour le protocole *Jabber*
- ▶ fonctionnement sur le **modèle client-serveur** :
 1. Juliette@Capulet.org envoie un message à Roméo@Montaigu.net
 2. le message de Juliette est traité par le serveur Capulet.org
 3. ce serveur établit la communication avec le serveur Montaigu.net
 4. le message est transmis entre les serveurs
 5. il est livré à Roméo

Inconvénients du courrier électronique

- ▶ incompatibilité des moyens de communication
- ▶ problèmes des alphabets nationaux
- ▶ pas d'authentification
- ▶ pas d'accusé de réception
- ▶ confidentialité pas vraiment garantie
- ▶ possibilité de fraude
- ▶ envahissement par les courriers non sollicités
- ▶ manque de pratique de trop nombreux utilisateurs
- ▶ outils grand public encourageant des pratiques nuisibles (codification complexes et inadaptées)
- ▶ nécessité de règles de comportement adaptées

L'idée de courrier électronique

- ▶ **propriétés de base** :
 - ▶ communication de personne à personne
 - ▶ sans nécessité de présence simultanée des deux correspondants
 - ▶ laisse une trace écrite
 - ▶ très grande vitesse de transmission
- ▶ **inconvénients** des autres moyens :
 - ▶ téléphone : pas de trace, coût, nécessité de simultanéité
 - ▶ courrier postal : lenteur, manque de fiabilité
 - ▶ groupes de discussion : beaucoup de bruit pour pas grand'chose, pas de confidentialité
 - ▶ protocole IRC : comme le précédent, plus nécessité de simultanéité et brièveté des messages
 - ▶ protocole HTTP : toute l'activité est du côté du client, donc pas de dialogue

Outils mis en jeu

- ▶ la **gestion normale** met en jeu deux types d'outils :
 - ▶ l'**agent de transport** effectue la transmission des messages d'une machine à une autre
 - ▶ l'**agent de courrier** sert à composer et envoyer des messages, et à les recevoir et afficher
- ▶ on utilise parfois les navigateurs pour simuler ce fonctionnement (bureau virtuel)

Adresse électronique

- ▶ les **mécanismes d'adressage** vus précédemment permettent d'identifier un ordinateur
- ▶ pour **identifier un utilisateur**, on combine son nom d'utilisateur à l'adresse d'un ordinateur dépositaire du courrier
- ▶ exemples :
 - ▶ ol@unice.fr
 - ▶ rms@prep.ai.mit.edu
- ▶ dans le premier cas, il n'y a qu'un nom de domaine
- ▶ dans le deuxième cas c'est un nom de machine précise
- ▶ des **mécanismes d'alias** gérés par les serveurs de courrier des domaines permettent plusieurs formes d'adresses :
 - ▶ Olivier.Lecarme@unice.fr
 - ▶ ol@echo.unice.fr
 - ▶ ol@pierredelune.i3s.unice.fr

L'agent de transport

- ▶ soit un message de toto@alpha.unice.fr à tutu@mit.edu
 1. l'agent de transport de la machine alpha.unice.fr doit **recevoir** le message préparé par Toto, identifiant clairement l'expéditeur et le destinataire
 2. il se **met en communication**, directement ou non, avec l'agent de transport de la machine réceptrice du domaine mit.edu
 3. il fournit ses références et celles de l'expéditeur, puis celles du destinataire
 4. il **transmet le message**
 5. il **termine la communication**

Le protocole SMTP

- ▶ **protocole** s'appuyant sur TCP et IP
- ▶ utilisant systématiquement le **port 25**
- ▶ exemple de communication :

```
telnet aiguemarine.i3s 25
Trying 134.59.133.66...
Connected to aiguemarine.i3s.unice.fr.
helo uranie
250 aiguemarine.i3s.unice.fr Hello uranie.unice.fr [134.59.2.99], pleased to
meet you
mail from : ol
250 ol... Sender ok
rcpt to : ol
250 ol... Recipient ok
data
354 Enter mail, end with "." on a line by itself
Essai
.
250 KAA00826 Message accepted for delivery
quit
221 aiguemarine.i3s.unice.fr closing connection
Connection closed by foreign host.
```

Complications

- ▶ la **communication directe** est souvent impossible
- ▶ pour beaucoup de domaines, un *serveur de courrier* centralise toutes les entrées et sorties
- ▶ il faut donc passer par des **relais de courrier**, qui transforment les adresses avant de retransmettre les messages
- ▶ l'**agent de transport** joue donc un rôle complexe et crucial :
 - ▶ il reçoit les requêtes des agents de courrier locaux
 - ▶ est à l'écoute sur le port 25 pour recevoir les requêtes d'autres agents de transport
 - ▶ il analyse les en-têtes des messages pour y chercher des informations
 - ▶ il dépose les messages dans les *boîtes aux lettres* des utilisateurs locaux
 - ▶ il ré-expédie certains messages à d'autres serveurs
 - ▶ il rejette certains messages
- ▶ l'utilisateur normal ne communique **jamais directement** avec l'agent de transport

L'agent de courrier

- ▶ l'*agent de courrier* a lui aussi un **rôle double** :
 - ▶ préparer un message et l'envoyer (par l'intermédiaire de l'agent de transport)
 - ▶ recevoir les messages
- ▶ les **messages** reçus sont rangés par l'**agent de transport** dans une *boîte aux lettres*
- ▶ les boîtes aux lettres gérées par un **serveur de courrier** sont des fichiers portant le **nom d'utilisateur du destinataire** et rangés dans un répertoire spécifique (par exemple `/var/spool/mail`)
- ▶ l'**agent de courrier** :
 - ▶ examine le contenu de la boîte aux lettres de l'utilisateur
 - ▶ en extrait les messages un à un (problème d'interblocage)
 - ▶ éventuellement, les archive dans les fichiers de l'utilisateur



Quelques commandes simples

- ▶ les **messages au départ** sont rangés dans une **file d'attente** pour être traités par l'agent de transport
- ▶ la commande `mailq` permet de **vérifier l'état** de cette file
- ▶ **en réception**, il est utile de savoir si l'on a du courrier en attente :
 - ▶ le shell l'indique à chaque nouvelle connexion
 - ▶ la commande `from` (pas toujours disponible) permet de lister les expéditeurs des messages non lus
 - ▶ le client X `xbiff` :
 - ▶ affiche l'icône d'une boîte aux lettres à l'américaine
 - ▶ quand du courrier est en attente, l'icône passe en vidéo inverse
 - ▶ chaque arrivée de message est signalée par un signal sonore
 - ▶ Emacs signale la présence de courrier en attente par le mot `Mail` (ou une icône) dans la barre de mode des fenêtres



Accès au courrier à distance

- ▶ la **boîte aux lettres** de l'utilisateur peut être **conservée sur le serveur de courrier**, sur lequel on ne se connecte normalement pas
- ▶ les **protocoles POP ou IMAP** permettent :
 - ▶ de consulter le courrier à distance, sans l'amener localement
 - ▶ éventuellement, de l'extraire de la boîte aux lettres et de le conserver sur la machine choisie
 - ▶ cela permet à des utilisateurs « nomades » d'accéder à leur courrier depuis n'importe quelle machine
 - ▶ cependant il faut un mot de passe transmis en clair
 - ▶ si les messages restent sur le serveur, ils occupent un **espace commun** et par conséquent très limité



Structure d'un message

- ▶ un *message* a la **structure générale** :
 - ▶ ensemble d'**en-têtes**, de forme normalisée
 - ▶ **corps du message**, éventuellement divisé en plusieurs parties
- ▶ certains en-têtes sont **définis par l'utilisateur** :
 - ▶ l'agent de courrier les définit d'emblée
 - ▶ l'utilisateur doit remplir certains champs
- ▶ d'autres en-têtes sont entièrement **préparés par l'agent de courrier**
- ▶ beaucoup d'en-têtes sont **ajoutés par les différents agents de transport**



En-têtes définis par l'utilisateur

- ▶ **To :**
 - ▶ indique le **destinataire**
 - ▶ placé par l'utilisateur, **obligatoire**
 - ▶ adresse électronique ordinaire, ou bien
 - ▶ "*chaîne quelconque*" <adresse>
 - ▶ plusieurs adresses séparées par des virgules
- ▶ **Subject :**
 - ▶ indique le **sujet du message**
 - ▶ **facultatif** mais très utile
 - ▶ la plupart des outils de courrier affichent ce champ dans la **liste des messages**
- ▶ **Cc :**
 - ▶ liste des **destinataires secondaires** (« copie carbone »)
 - ▶ **facultatif**, pas toujours demandé par l'agent de courrier

Champs ajoutés par l'agent de courrier

- ▶ **Date :**
 - ▶ date et heure exacte d'**expédition**
 - ▶ le **fuseau horaire** est indiqué (code et décalage par rapport au temps universel)
- ▶ **Return-path :**
 - ▶ indique comment **renvoyer le message** s'il ne peut pas être transmis à son destinataire
 - ▶ normalement identique au champ **From :** ou au champ **Sender :**
- ▶ **In-reply-to :**
 - ▶ ajouté si le message **répond à un autre**
 - ▶ indique l'**identification du message** et de l'expéditeur
 - ▶ utilise le champ **Message-id :** construit par l'agent de courrier
- ▶ **Received :**
 - ▶ champ apparaissant **plusieurs fois**
 - ▶ montre en **ordre chronologique** le chemin suivi par le message

Autres en-têtes

- ▶ **From :**
 - ▶ indique l'**expéditeur**
 - ▶ **obligatoire**, introduit par l'agent de courrier
 - ▶ si l'utilisateur le **modifie**, alors l'agent de courrier doit ajouter le champ **Sender :** pour indiquer l'**expéditeur effectif**
- ▶ **Bcc :**
 - ▶ liste des **destinataires cachés**
 - ▶ **facultatif**, demandé par l'utilisateur
 - ▶ ces destinataires reçoivent le message, mais n'**apparaissent pas** dans la liste des destinataires

Codage MIME

- ▶ les **en-têtes** du message doivent n'utiliser qu'un **alphabet sur 7 bits** (code ASCII)
- ▶ les **caractères accentués** (par exemple) sont **transcodés** dans les en-têtes
- ▶ le **corps du message** est limité à un **alphabet sur 8 bits** (par exemple ISO-8859-1)
- ▶ certains **agents de courrier** ou de **transport** préfèrent le limiter à un alphabet sur 7 bits
- ▶ le codage *quoted-printable* code sur 7 bits les caractères sur 8
- ▶ le **codage MIME** (*Multipurpose Internet Mail Extension*) permet de :
 - ▶ utiliser plusieurs alphabets dans le même message
 - ▶ inclure des fichiers de tous types (« fichiers attachés »)
 - ▶ enrichir le message par des indications de présentation
- ▶ les agents de courrier simples n'interprètent pas tout cela

Conseils de bonne utilisation du courrier

- ▶ **ne codez jamais** les messages en HTML !
 - ▶ c'est inutile, lourd et encombrant
 - ▶ ce codage n'est pas fait pour cela
 - ▶ bien des agents de courrier n'en veulent pas
 - ▶ cela dénote un expéditeur béotien
- ▶ **respectez** les mêmes règles de politesse, bonne présentation, concision, orthographe que pour un **vrai courrier**
- ▶ n'envoyez vos messages qu'à ceux qui sont concernés

Conseils de bonne utilisation du courrier (suite)

- ▶ n'abusez pas de l'humour, et signalez-le
- ▶ ne comptez pas sur une confidentialité totale
- ▶ ne **ré-expédiez** pas à quelqu'un un message qui ne lui est **pas destiné**
- ▶ mettez toujours un sujet explicite, c'est peut-être la seule chose qui sera vue par le destinataire!
- ▶ si vous répondez à un message et incluez celui de votre correspondant :
 - ▶ faites-le seulement si c'est nécessaire
 - ▶ distinguez clairement ce qui est de lui et ce qui est de vous
- ▶ ne sous-estimez pas les délais ordinaires de réponse, ou de notification d'erreur : ne renvoyez pas trois fois le même message dans une heure