Le protocole TCP

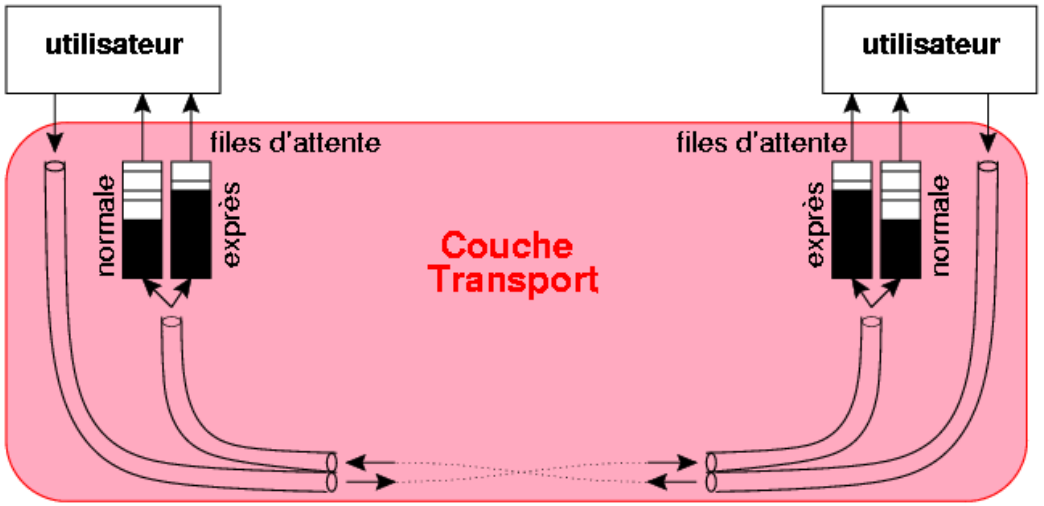
Généralité

Port de la connexion

P. Laurençot

TCP

- mode connecté → 3 phases : **connexion, échange , déconnexion**
- établissement de connexion TCP = création des 2 canaux
  - mise en place de files d'attente des octets en attente de livraison
- terminaison de connexion = destruction des canaux



Point d'accès aux services TCP/UDP

• Le concept de point d'accès au Service TCP (resp. UDP) est traduit par le concept **de socket** des systèmes d'exploitation

• **La socket** : lien entre l'applicatif et la couche transport

•*Une socket par communication.*

❑ création de *socket* selon les besoins des processus

- par la fonction **`socket()`**
- **une *socket* appartient à un processus**

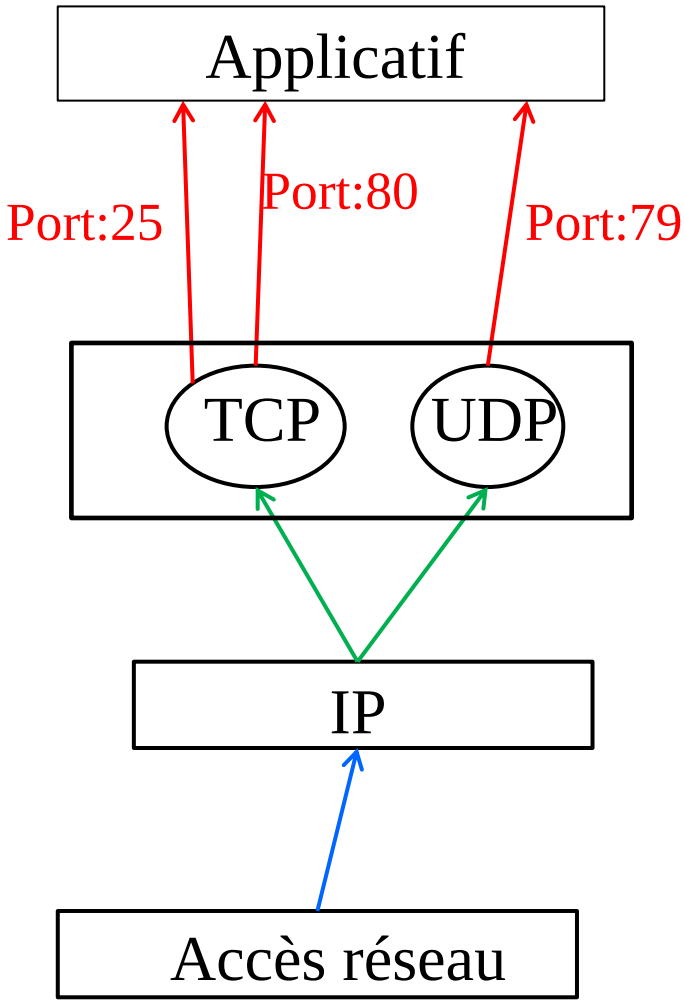
❑ Création du lien couche 4/ couche applicative via numéro de port

- par la fonction **`bind()`** (implicite ou explicite)

Port d'une connexion (1)

- Identifier un processus via une socket
  - 3 informations obligatoires :
    - L'adresse IP
    - Le protocole utilisé
    - Le numéro de port
      - Privilégié
      - Bien connu/enregistré

Certaines informations sont implicites ou configurables pour certaines applications



Port d'une connexion (2)

- Port applicatif
  - Unicité d'un port / applicatif
  - Privilégié (<1024) → seul l'administrateur peut utiliser ces ports
    - 22 : ssh , 23 : telnet , 80 : http, 110 : POP3, 143: IMAP…
  - enregistré >= 1024
    - 8080 : alternate http, 6000 : serveur X, 3724 : warcraft,..
  - Dynamique ou privé >= 49152

Transmission de données

Les données sont transmises dans des PDU-TCP (resp.PDU-UDP), transportées par le Service IP qui livre ... peut-être, une seule fois

Les risques :

- perte de PDU-IP → perte de la PDU-TCP ou PDU-UDP encapsulée et des données qu'elle contient
- livraisons multiples de la même PDU (duplication), donc de données
- livraison d'une suite d'octets dans un ordre différent de celui de la soumission

UDP ne fait ni détection, ni correction

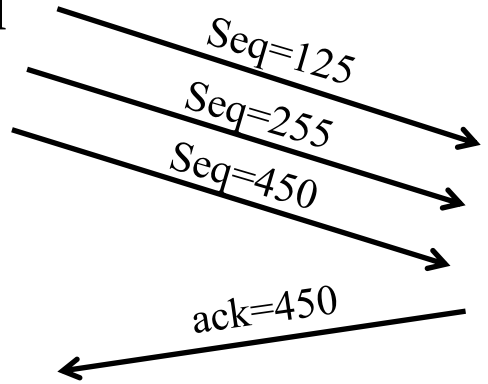
TCP tente de détecter tous les problèmes et de les corriger

Fiabilité de TCP (1)

- Numérotation lors de l'expédition
  - les octets de l'utilisateur sont transmis par bloc dans des PDU-TCP
  - l'entité-TCP calcule un numéro de séquence qui correspond "aux nombres d'octets envoyés" et le place dans la PDU envoyé, cela correspond au numéro seq : séquence
  - initialisation de seq : à l'établissement de connexion
  - Calcul lors de la phase d'initialisation d'un nombre correspondant "aux nombres d'octets reçus" (aussi envoyé dans la PDU), c'est le numéro ack : acquittement
- Vérification lors de la réception
  - n° reçu > n° attendu → une perte
  - n° reçu < n° attendu → une duplication
  - n° reçu = n° attendu → OK
- Correction
  - *duplication* : l'entité-TCP ignore la PDU-TCP en doublon
  - *perte* : retransmission de la PDU contenant le bloc
    - on renvoie la partie perdue

Fiabilité de TCP (2)

- Différence paquet perdu /paquet en retard
  - TCP peut acquitter plusieurs segments d'un coup
    (gain en bande passante)
  - Mise en place de **timers** (temporisation)
    - A chaque émission, création d'un timer
      - Si ack avant fin timer ⇒ OK
        sinon segment perdu



- Calcul du timer
  - Si timer trop petit → beaucoup de retransmission
  - Si timer trop grand → attente longue pour détecter une perte
  - Re-Calcul du timer de temps en temps
    - basé sur le temps pour un échantillon, le délai moyen, la variance, etc…
      (timer >> RTT)

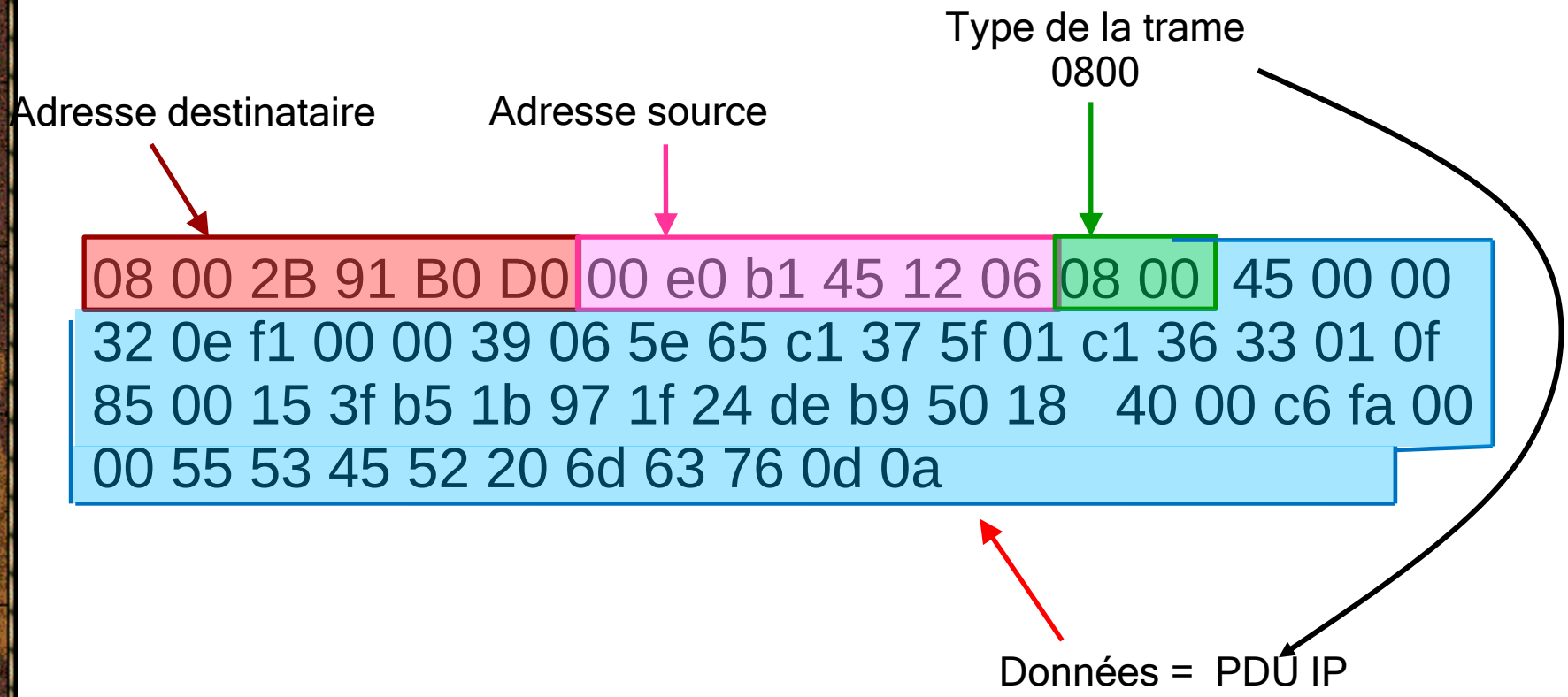
Protocole TCP - contrôle de flux

- Une entité-TCP place les octets reçus pour l'utilisateur dans des files d'attente où il peut en prendre livraison
  - place dans la file d'attente = capacité de réception
  - file d'attente pleine
    ⇒ impossible d'ajouter de nouveaux octets reçus
    et consommation inutile de ressources de (re)transmission

- Le contrôle de flux permet d'éviter à l'entité-TCP expéditrice d'envoyer plus que l'entité-TCP réceptrice peut mémoriser dans les files d'attente
  - chaque entité-TCP place dans chaque PDU qu'elle envoie le nb d'octets libres dans la file d'attente (champ win – tcp window size)

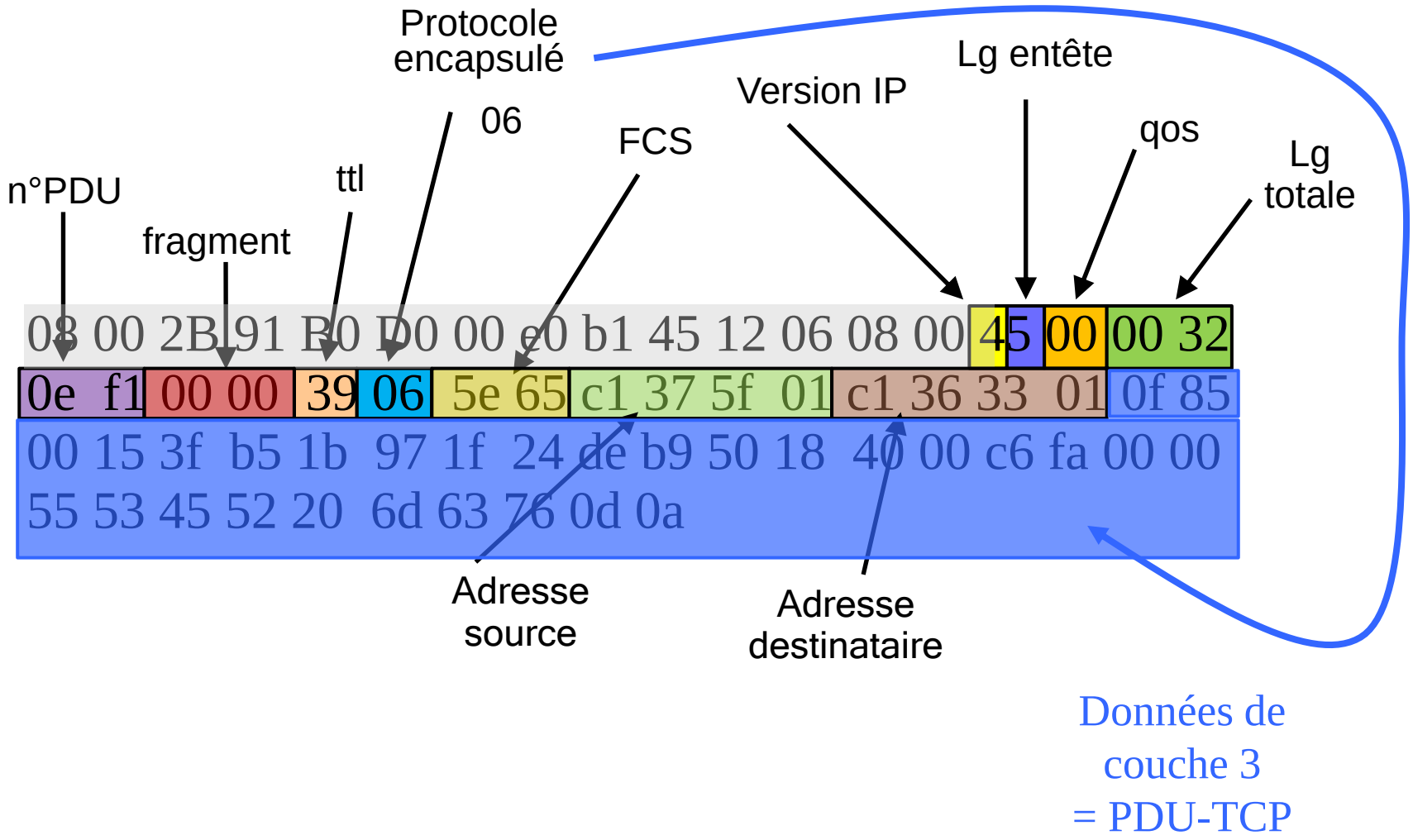
Exemple de trame

1) la PDU MAC-Ethernet

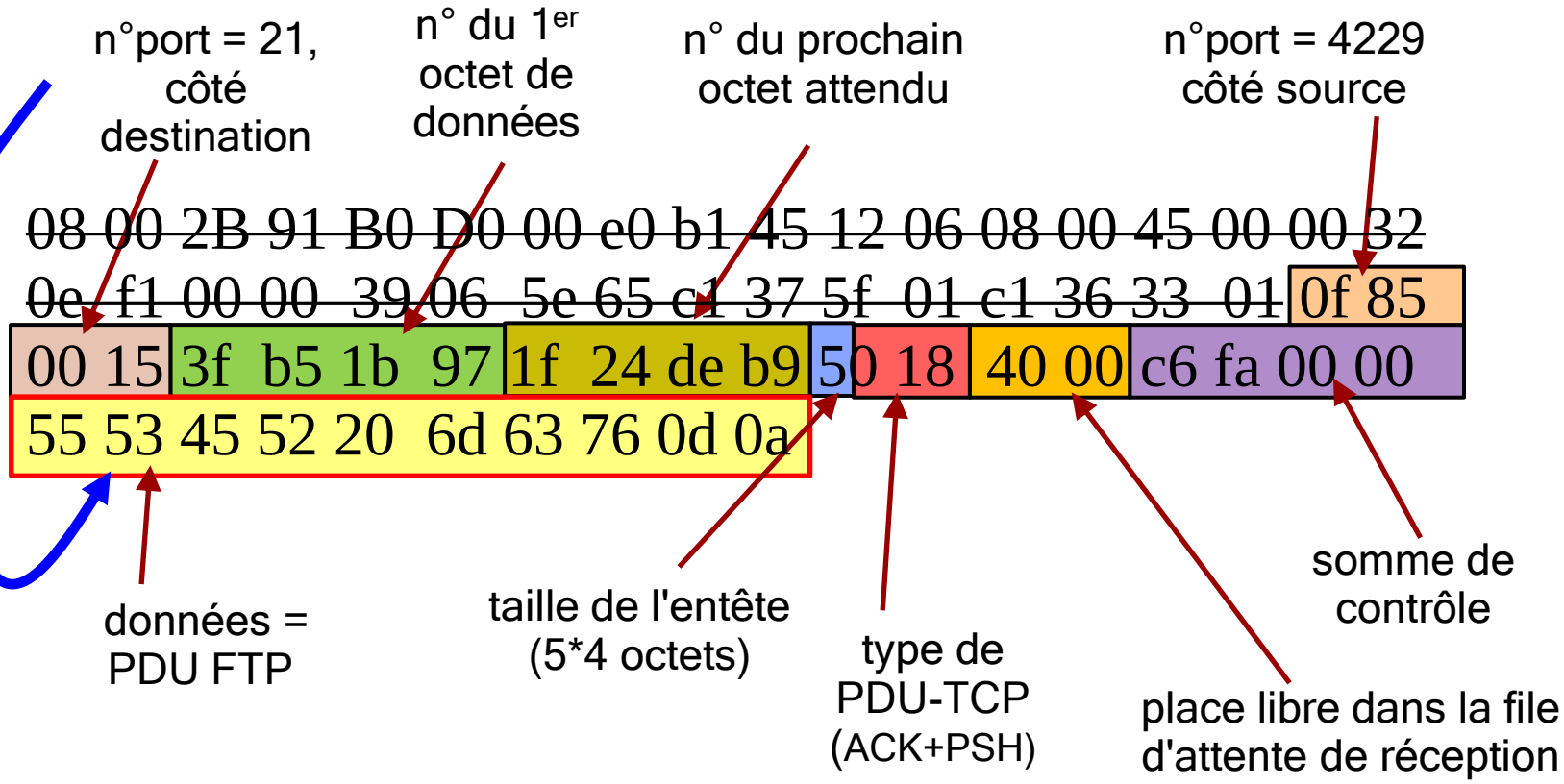


Exemple de trame

2) la PDU IP



Exemple de PDU-TCP



type de PDU-TCP

URG	ACK	PSH	RST	SYN	FIN
20	10	8	4	2	1

ASCII : 55->U, 53->S, 45->E, 52-> R, 6d->m

La couche Applicative
et
quelques applications

Généralités

Différents protocoles applicatifs

P. Laurençot

La couche Applicative
Généralités

Les Services d'Application

Un Service d'Application est conçu pour

- *rendre accessibles des informations mémorisées sur un système distant*

courrier électronique	transfert de fichiers
web	partage d'informations d'administration

- *rendre accessible une ressource appartenant à un système distant*

ressource de calcul	travail à distance
ressource logiciel	partage d'imprimantes
ressource en disque	partage de fichiers

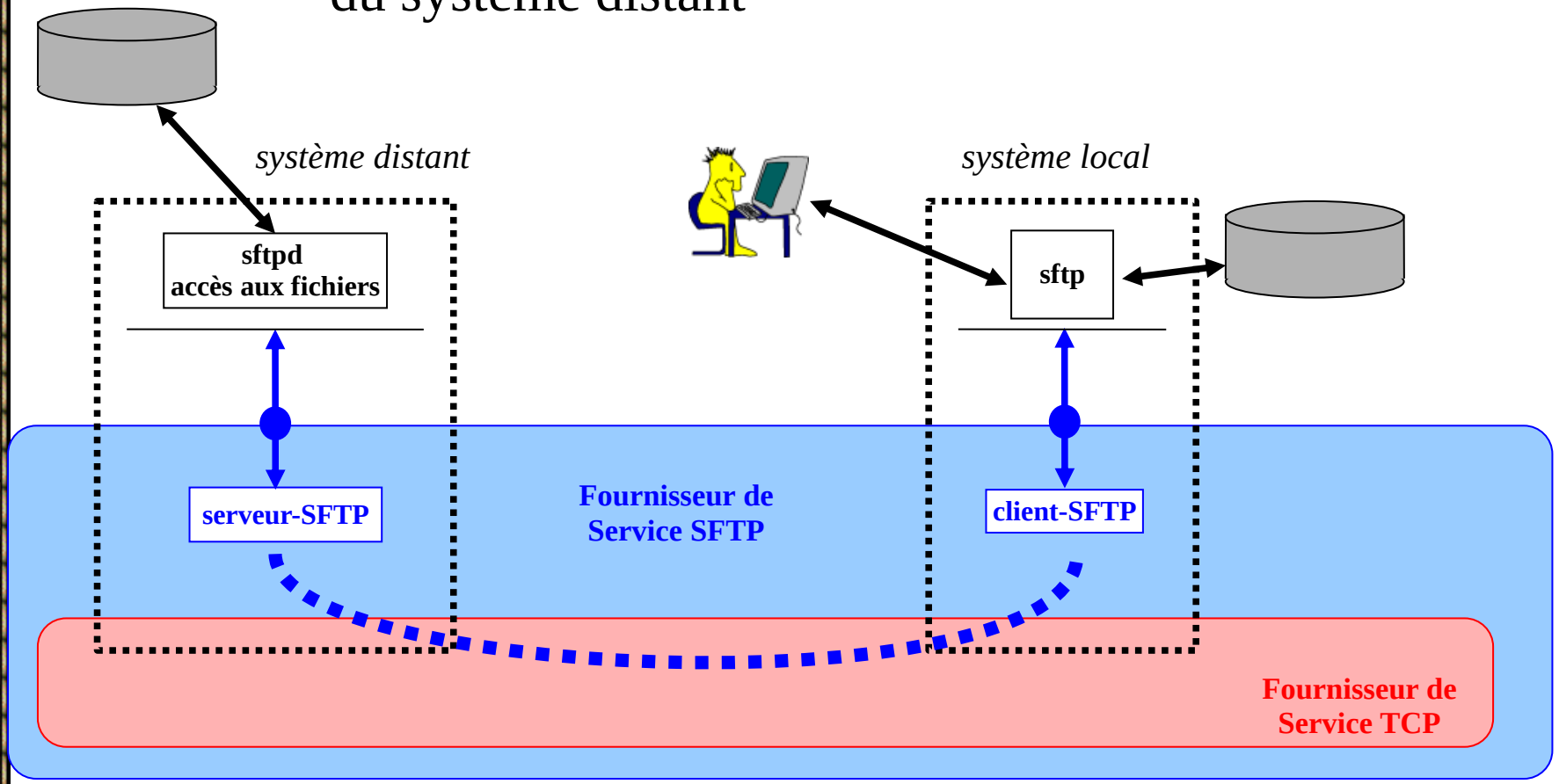
une couche Application par Service

Organisation d'un fournisseur de Service d'Application

- sur le système qui possède une information ou une ressource : une entité-serveur
- sur le système qui a besoin d'une information ou d'une ressource : une entité-client
- la transmission des PDU entre client(s) et serveurs(s) est effectuée par l'utilisation d'un Service inférieur

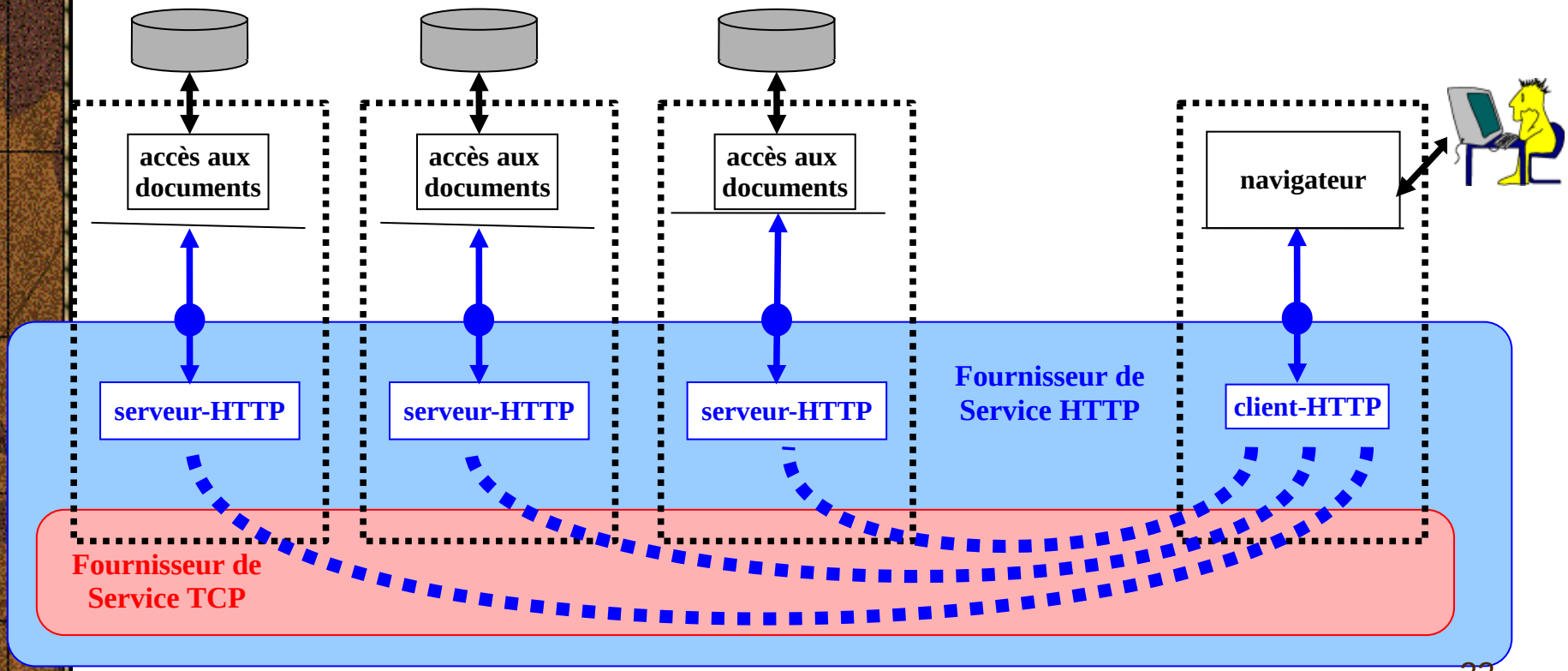
Service de Transfert de fichiers (SFTP) : 1 serveur – 1 client

- le processus qui s 'exécute sur le système local a accès, en même temps, à ses disques et aux disques du système distant



Service HTTP (Web) : N serveurs – 1 client ou 1 serveur – N clients

- 1 navigateur – client HTTP – demande des documents à un ou plusieurs serveurs
- et chaque serveur-HTTP peut servir plusieurs navigateurs



Généralité

Notion client/serveur

- Serveur : en attente de connexion → un port ouvert
- Client : veut une information détenue par le serveur
  - ***Connait le nom ou l'@IP du serveur***
  - Le numéro de port
  - La requête à effectuer

Client

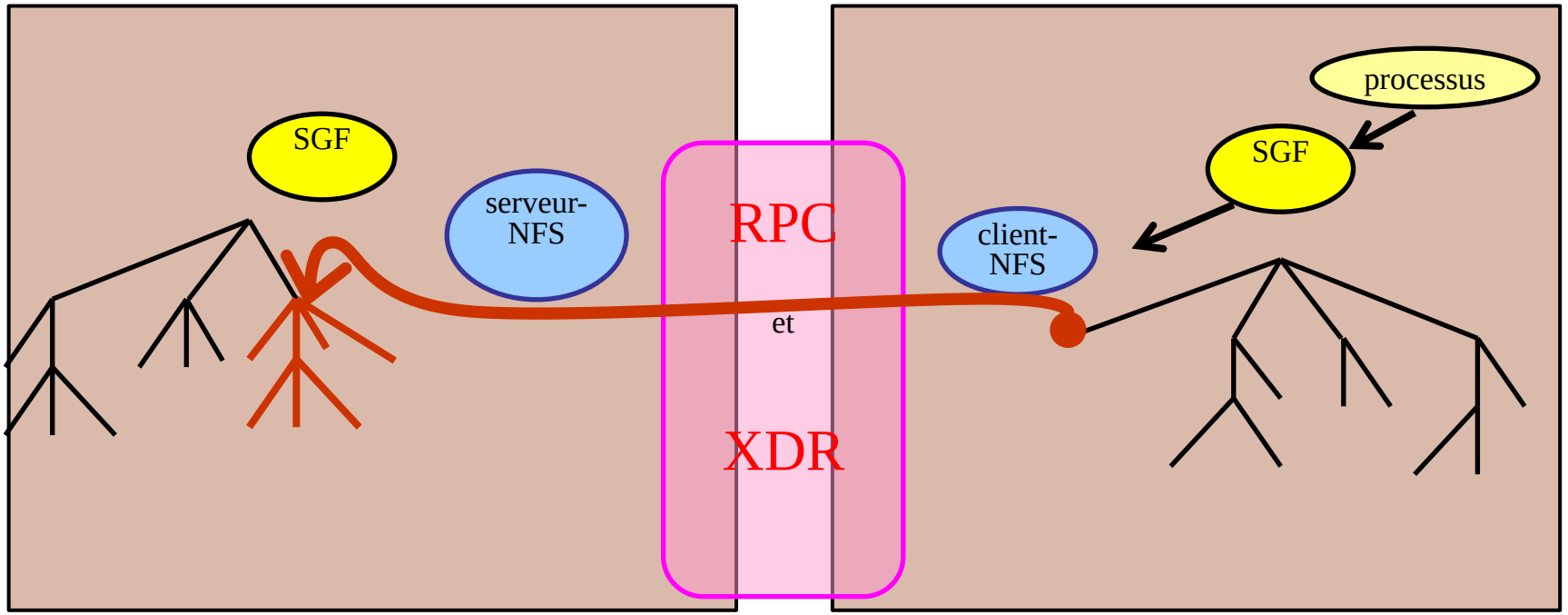
- processus créé lorsque le besoin apparaît
  - par une personne
  - au démarrage du système
- l'entité-client doit savoir à quelle entité-serveur elle doit/peut s'adresser
  - au lancement du processus : fichier de configuration, argument
  - par saisie pendant l'exécution

Généralité - Côté serveur

- Les entités serveur
  - fonctionnement permanent (démons)
    - processus créé au démarrage du système
  - processus lancé lorsqu'une demande arrive
    - par un processus "écouteur"
- Méthodes de prise en charge d'une demande de Service
  - L'entité serveur peut traiter elle-même la demande
    - Mise en attente des autres demandes jusqu'à la fin du traitement
  - L'entité serveur peut déléguer à un processus fils (ou un thread) le traitement de la demande
    - Contrôle du nombre de fils (ou de thread) ->nombre maximum donné, pré-création …
    - Risque de surcharge de la mémoire

Quelques exemples
d'Applicatifs

NFS (1) : partage de répertoires



serveur-NFS	facilités	client-NFS
`mountd`	montage	`mount`
`nfsd`	opérations sur les fichiers et répertoires	`nfsiod`

NFS (2) : caractéristiques

- *identifications*
  - du serveur, du client :
    - par l'adresse-IP des systèmes
  - des utilisateurs : par UID-GID

- serveur-NFS sans état

- *caches des clients-NFS*
  - attributs des fichiers et répertoires
  - lectures anticipées
  - écritures retardées

- *autorisations de montage*
  - liste des répertoires exportés
  - liste des clients autorisés

- *autorisations d'accès aux fichiers*
  - selon UID-GID
  - UID=0 (root) devient UID=65534 (nobody) sauf configuration

- Service support : RPC

  (Remote Procedure Call)
  - liste de fonctions :

```
getattr   read      remove
setattr   write     rename
lookup    create    ...
```

  - paramètres :

    codés par XDR *(eXternal Data Representation)*
  - service support de RPC :
    - UDP ou TCP

NFS (3) : configuration

Serveur

- construire la liste des répertoires exportables `/etc/exports`
  - options
    - clients autorisés
    - type de montage (ro, rw)
    - contrôle des UID-GID
- vérifier que le Service RPC est lancé `rpcinfo -u localhost`
- lancer les serveurs
  - de montage
  - d'opérations sur les fichiers
  - de verrouillage des fichiers

  `/etc/init.d/nfs start`

  `service nfs-kernel-server start`
- vérifier l'exportation

Client

montage

- à partir d'un fichier de configuration
- manuellement
- options de montage habituelles ou spécifiques à NFS (voir manuel `mount(8)`)

opérations sur les fichiers

- par les fonctions habituelles, transparents pour les utilisateurs

Informations d'administration

- informations nécessaires pour le fonctionnement des systèmes
  - identification des utilisateurs — comptes, mots de passe, adresses-mail, ...
  - identifications des systèmes — noms, adresses-MAC, adresses-IP,…
  - identification des entités du réseau — appellations des protocoles, des applications …
- domaine d'administration

  ensemble des systèmes qui ont besoin des mêmes informations d'administration

  - **des bases d'informations d'administration**
    - base des comptes — base des adresses-mail
    - base des systèmes — ...
  - **un ou plusieurs serveurs d'informations d'administration**
- partage d'informations d'administration
  - primitives d'accès aux informations d'administration
  - réglementation des accès

les solutions les plus connues		
NIS	Network Information System	SUN
DCE	Distributing Computing Environment	ISO
SMB	System Message Block	Microsoft
LDAP	Lightweight Directory Access Protocol	IETF
AD	Active Directory	Microsoft

NIS (1) : définitions

domaine-NIS

défini par un nom (de domaine)
chaque système doit savoir à quel domaine-NIS, il appartient

map-NIS

un ensemble d'informations d'administration de même type
une map est construite à partir d'une source (fichier texte, ou autre forme)
exemple : la map `passwd.byname` est construite à partir de la source : `/etc/passwd`
copie maître, copies esclaves

serveur-NIS

un serveur-NIS maître par domaine, il possède la copie maître des maps.
un ou plusieurs serveur-NIS esclaves par domaine, ils possèdent une copie esclave des maps

installé sur un système qui possède une copie des maps
il répond aux demandes d'accès aux maps

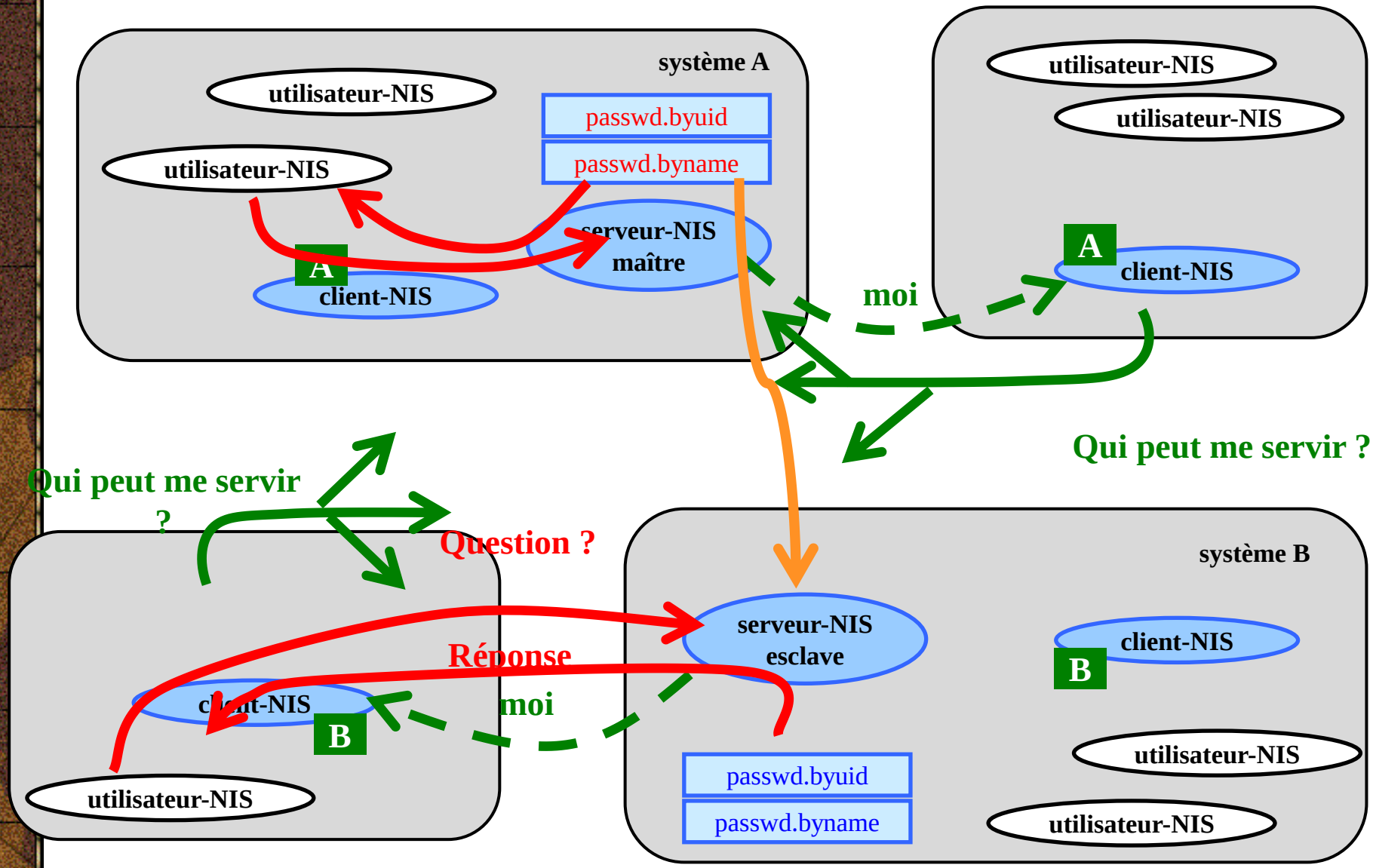
client-NIS

installé dans tout système du domaine qui utilise des informations d'administration fournies par le Service-NIS

utilisateur-NIS

entité qui a besoin d'accéder à des informations d'administration

NIS (2) : fonctionnement



DHCP : définitions

DHCP : Dynamic Host Configuration Protocol -> RFC 2131

(successeur de BOOTP)

utilisation de la couche transport UDP : port 67 client ->serveur

port 68 serveur -> client

• Utilisation pour récupération automatique @IP, etc… sur un ordinateur

Serveur fournit (en général) :

- Adresse IP
- Masque de réseau
- Adresse de la passerelle
- Adresse du DNS, nom du domaine

Un serveur a un « pool » d'adresses qu'il peut distribuer

➡ utilisation d'un bail sur la durée de location d'une adresse

DHCP : fonctionnement

1. Emission **d'un broadcast** pour la demande d'une adresse IP
   ->DHCPDISCOVER
2. Réponse en unicast d'un serveur vers le client en lui spécifiant les différents paramètres à utiliser
   (auparavant le serveur fait 2 pings pour savoir si l'adresse qu'il propose n'est pas déjà utilisée…)
   ->DHCPOFFER
3. Client donne son accord sur cette adresse en faisant un broadcast
   -> DHCPREQUEST
4. Le serveur acquitte cette accord en répondant en unicast
   -> *DHCPACK*

Pour libérer une adresse IP, utilisation de *DHCPRELEASE*

Utilisable que sur le réseau LAN (par défaut)

Abandonné pour IPv6, et remplacé par ICMPv6

ICMP

- *ICMP : Internet Control Message Protocol*
  - Permet le contrôle des erreurs de transmission
  - Composé principalement :
    - D'un entête IP
    - D'un type de message
  - Différent type de message :
    - Type 0 : echo réponse
      → réponse au type 8
    - Type 3 : destinataire inaccessible
    - Type 5 : redirection
    - Type 8 : demande echo
      → permet de savoir si une @IP répond (ping)

Autres protocoles applicatifs

- Quelques noms:
  - NTP : network Time Protocol
    - Permet de synchroniser les machines à l'heure universelle
  - SNMP : Simple Network Management Protocol
    - Permet la gestion du réseau en récupérant des indicateurs sur les équipements
  - SIP : Session Initiation Protocol
    - Protocole de signalisation utilisé dans la VoIP, vidéoconférence,…
  - RTSP : Real Time Streaming protocol
    - Permet de contrôler la transmission audio/video entre entités
  - FTP/TFTP : Trivial File Transfert Protocol
  - RDP : Remote Desktop Protocol (port 3389)
  - SMTP, IMAP, POP, HTTP, DNS….

Service de messagerie

- Autre nom : courrier électronique, e-mail, couriel
  - ➡ permet d'échanger des messages et des fichiers

- Il nécessite un serveur de messagerie accessible à partir d'internet. Le serveur dispose d'une boîte à lettre (BAL) pour chaque client géré par la messagerie.

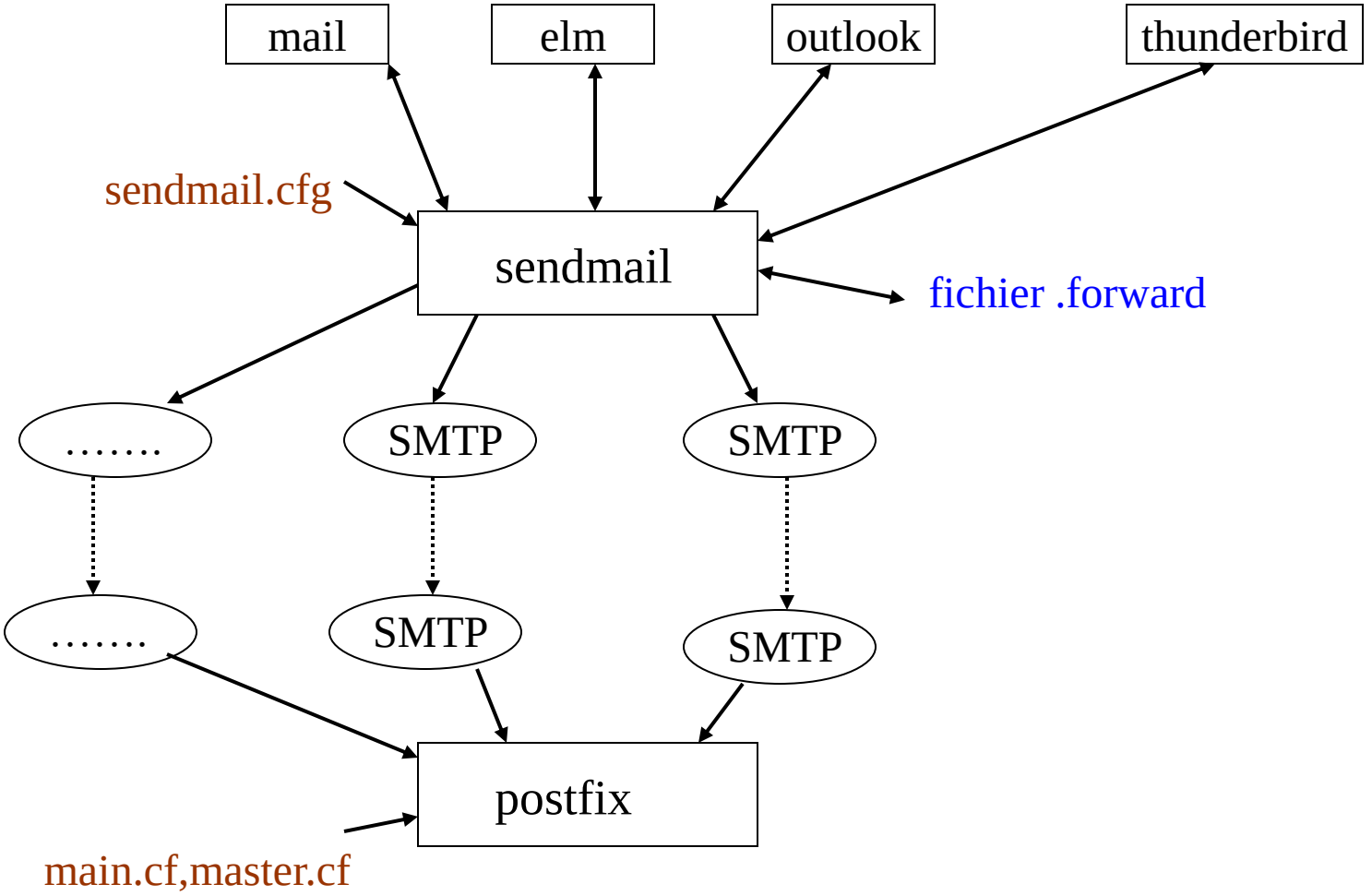
- Les messages sont stockés par le serveur de messagerie, en attendant que le client vienne consulter sa boîte aux lettres.

  - le message peut être lu :
    - en *mode online* - message stocké sur le serveur et lu à distance
    - en *mode offline* – message déplacé sur la station client et effacé du serveur

Terminologie du mail

- Logiciels de messagerie (user agent)
  - Thunderbird, Eudora, Lotus Notes, Outlook, Mail, elm, xmh, …
- Logiciels de gestion de messagerie (**M**essage **T**ransfert **A**gent)
  - Exchange server, Domino mail server, sendmail, exim, postfix…
  - Hybride : Zimbra
- Protocole de transport des messages (couche applicative)
  - SMTP, UUCP, X400, X500,…
- Stockage des messages
  - /usr/spool/mail/…, /var/mail/…

Rôle des entités du mail



SMTP

- SMTP: RFC 821

  Simple Mail Transfert Protocol

  - Permet d'échanger du courrier électronique

- le format des adresses des utilisateurs fait figurer le nom de l'utilisateur suivi du nom de domaine : laurenco@isima.fr

  identifie de manière unique chaque boîte aux lettres (personne@machine.domaine, personne@domaine)

  (cf. requête MX du DNS)

- le codage utilisé pour le message et les fichiers attachés :
  - texte pur codé en ASCII 7 ou 8 bits (RFC 822) pour une prise en compte des caractères accentués
  - standard MIME (Multipurpose Internet Mail Extension) pour du texte formaté, des images ou du son

RFC 822

Structure d'un message

- En tête
- Ligne blanche
- Corps du message (suite de lignes terminés par CR/LF)

En tête

- From : expéditeur
- To : destinataire(s)
- CC : copies à
- Bcc : copie aveugle (destinataire caché)
- Reply-to : adresse de réponse
- Error-to : adresse en cas d'erreur
- Date : date et heure de l'envoi
- Message-id : numéro unique permettant de référencer le message
- Received : informations de transfert
- Subject : sujet

Structure d'un courrier

- Limitations
  - Tout est sous forme de lignes ASCII
  - 2 parties
    - l'entête **(définit les services attendus)**
    - corps **(le texte de la lettre est terminé par une ligne avec "." comme premier et unique caractère)**
  - nom d'utilisateur < 64 caractères
  - nom de domaine < 64 caractères
  - nombre de destinataires < 100
  - une ligne < 1000 catactères
  - utilisation du format MIME (RFC 2045) pour envoyer des fichiers non ASCII

Session SMTP (1)

- 3 phases
  - Etablissement de la connexion au niveau smtp et identification de la source et la destination
  - Envoi du message avec les différents entêtes
  - Libération de la connexion

- Session
  - Connexion tcp sur le port 25
  - Le serveur renvoie le code 220 (service disponible) suivi de son nom

    ex : 220 sp.isima.fr SMTP sendmail….
  - Envoi d'une requête de connexion par la commande : *HELO*
  - Réponse par le code 250 (OK), et nom, please to meet you

    la connexion au niveau SMTP est réalisé

Session SMTP (2)

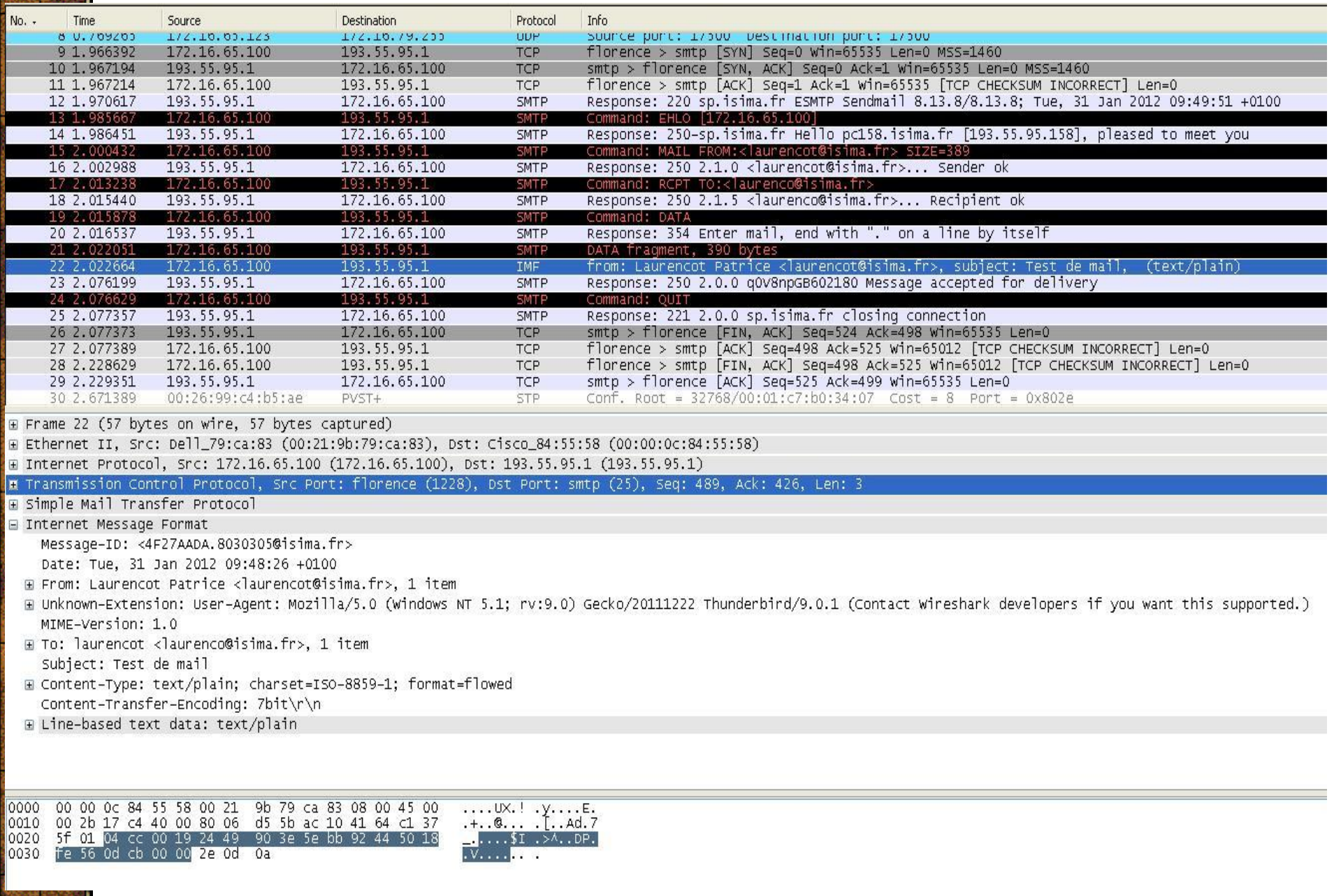
- Session (suite)
  - Envoi du nom de l'expéditeur, réponse 250 (OK) *(MAIL FROM)*
  - Envoi du nom du destinataire, réponse 250 (OK) *(RCPT TO)*
  - Début du transfert du message *(commande DATA)*
    réponse : *354 Enter mail, end with "." on a line by itself\r\n*
  - Envoi du message : Message-id, From, To, Date, Message, …
  - Réponse : Message accepted for delivery
  - Pour terminer : QUIT
    réponse : 221 sp.isima.fr closing connection

- Les "Received" indiquent le chemin suivi, dans l'ordre inverse
  - ils sont ajoutés par les machines (relais SMTP) à travers lesquelles le message a transité
  - ils permettent de retrouver l'origine du message
  - cela évite le bouclage de messages (max 25 champs received)

ESMTP

- Plus récent que SMTP
- Apporte des fonctionnalités supplémentaires
  - taille des messages
  - transport des messages en 8 bits
  - plus de fonctions disponibles

  - Le message de bienvenue est *EHLO.* En cas de réponse négative, le client doit basculer vers l'ancien protocole.


No. Time Source Destination Protocol Info
8 0.769265 172.16.65.123 172.16.79.255 UDP Source port: 17500 Destination port: 17500
9 1.966392 172.16.65.100 193.55.95.1 TCP florence > smtp [SYN] Seq=0 Win=65535 Len=0 MSS=1460
10 1.967194 193.55.95.1 172.16.65.100 TCP smtp > florence [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
11 1.967214 172.16.65.100 193.55.95.1 TCP florence > smtp [ACK] Seq=1 Ack=1 Win=65535 [TCP CHECKSUM INCORRECT] Len=0
12 1.970617 193.55.95.1 172.16.65.100 SMTP Response: 220 sp.isima.fr ESMTP Sendmail 8.13.8/8.13.8; Tue, 31 Jan 2012 09:49:51 +0100
13 1.985667 172.16.65.100 193.55.95.1 SMTP Command: EHLO [172.16.65.100]
14 1.986451 193.55.95.1 172.16.65.100 SMTP Response: 250-sp.isima.fr Hello pc158.isima.fr [193.55.95.158], pleased to meet you
15 2.000432 172.16.65.100 193.55.95.1 SMTP Command: MAIL FROM:<laurencot@isima.fr> SIZE=389
16 2.002988 193.55.95.1 172.16.65.100 SMTP Response: 250 2.1.0 <laurencot@isima.fr>... Sender ok
17 2.013238 172.16.65.100 193.55.95.1 SMTP Command: RCPT TO:<laurenco@isima.fr>
18 2.015440 193.55.95.1 172.16.65.100 SMTP Response: 250 2.1.5 <laurenco@isima.fr>... Recipient ok
19 2.015878 172.16.65.100 193.55.95.1 SMTP Command: DATA
20 2.016537 193.55.95.1 172.16.65.100 SMTP Response: 354 Enter mail, end with "." on a line by itself
21 2.022051 172.16.65.100 193.55.95.1 SMTP DATA fragment, 390 bytes
22 2.022664 172.16.65.100 193.55.95.1 IMF from: Laurencot Patrice <laurencot@isima.fr>, subject: Test de mail, (text/plain)
23 2.076199 193.55.95.1 172.16.65.100 SMTP Response: 250 2.0.0 q0V8npGB602180 Message accepted for delivery
24 2.076629 172.16.65.100 193.55.95.1 SMTP Command: QUIT
25 2.077357 193.55.95.1 172.16.65.100 SMTP Response: 221 2.0.0 sp.isima.fr closing connection
26 2.077373 193.55.95.1 172.16.65.100 TCP smtp > florence [FIN, ACK] Seq=524 Ack=498 Win=65535 Len=0
27 2.077389 172.16.65.100 193.55.95.1 TCP florence > smtp [ACK] Seq=498 Ack=525 Win=65012 [TCP CHECKSUM INCORRECT] Len=0
28 2.228629 172.16.65.100 193.55.95.1 TCP florence > smtp [FIN, ACK] Seq=498 Ack=525 Win=65012 [TCP CHECKSUM INCORRECT] Len=0
29 2.229351 193.55.95.1 172.16.65.100 TCP smtp > florence [ACK] Seq=525 Ack=499 Win=65535 Len=0
30 2.671389 00:26:99:c4:b5:ae PVST+ STP Conf. Root = 32768/00:01:c7:b0:34:07 Cost = 8 Port = 0x802e
Frame 22 (57 bytes on wire, 57 bytes captured)
Ethernet II, Src: Dell_79:ca:83 (00:21:9b:79:ca:83), Dst: Cisco_84:55:58 (00:00:0c:84:55:58)
Internet Protocol, Src: 172.16.65.100 (172.16.65.100), Dst: 193.55.95.1 (193.55.95.1)
Transmission Control Protocol, Src Port: florence (1228), Dst Port: smtp (25), Seq: 489, Ack: 426, Len: 3
Simple Mail Transfer Protocol
Internet Message Format
Message-ID: <4F27AADA.8030305@isima.fr>
Date: Tue, 31 Jan 2012 09:48:26 +0100
From: Laurencot Patrice <laurencot@isima.fr>, 1 item
Unknown-Extension: User-Agent: Mozilla/5.0 (Windows NT 5.1; rv:9.0) Gecko/20111222 Thunderbird/9.0.1 (Contact Wireshark developers if you want this supported.)
MIME-Version: 1.0
To: laurencot <laurenco@isima.fr>, 1 item
Subject: Test de mail
Content-Type: text/plain; charset=ISO-8859-1; format=flowed
Content-Transfer-Encoding: 7bit\r\n
Line-based text data: text/plain
0000 00 00 0c 84 55 58 00 21 9b 79 ca 83 08 00 45 00 ....UX.! .y....E.
0010 00 2b 17 c4 40 00 80 06 d5 5b ac 10 41 64 c1 37 .+..@... .[..Ad.7
0020 5f 01 04 cc 00 19 24 49 90 3e 5e bb 92 44 50 18 _.....$I .>^..DP.
0030 fe 56 0d cb 00 00 2e 0d 0a .V.....

Format MIME (1)

- Types d'encodage
  - Texte 7 bits, ASCII
  - Texte 8 bits
    - les textes sont composés de caractères 8 bits
    - il faut préciser l'alphabet : ex : iso-latin1 (alias iso-8859-1)
  - Base 64
    - pour les messages binaires
    - groupe de 24 bits, segmentés en 6 bits
      (3 octets, 4*6 bits 0 – 25 → A – Z , 26 – 51 → a –z, 52 – 61 → 0 – 9, 62 → +, 63 → /)
  - Quoted-Printable
    - codage ASCII normal *(A =0x41, a= 0x61, 0=0x30, +=0x2B, /=0x2F)*
    - pour ce qui n'est pas ASCII, valeur = hexa.

Format MIME (2)

- Contenu est divisé en 7 types différents, eux-mêmes re-divisés en sous-types.
  - Text
    - plain, richtext
  - Image
    - Gif, jpeg
  - Audio
    - basic
  - Video
    - mpeg
  - Application
    - octet-stream (WORD), postscript, …
  - Message
    - rfc822, partial, external-body (référence à un fichier dans internet)
  - Multipart
    - mixed, alternative (plusieurs formats) , parallel, digest (plusieurs messages)

Format MIME (3)

- Exemple
  - From :
  - Mime-Version : 1.0
  - To :
  - Subject
  - Content-type : multipart/mixed; boundary=« debutdumime"
    (ligne blanche)

This is a multi-part message in MIME format.

--debutdumime
content-type : text/plain; charset=iso-8859-1

bla-bla
--debutdumime
content-type : audio/basic
content-transfert-Encoding: base64

fichier audio
--debutdumime

POP

- Protocole permettant de relever le courrier sur un serveur
  - POP 3 : Post Office Protocol (version 3) port 110
    - permet l'authentification (login, passwd en clair – codage bin64)
      (pops permet de sécuriser les échanges)
    - réception seulement des courriers sur un serveur (envoi par smtp)
    - réception des messages d'erreur ou d'acquittement
    - Nécessité de télécharger l'intégralité du courrier sur la station avant la lecture, sans possibilité de manipuler directement les messages sur le serveur
    - POP utilise une syntaxe en 4 caractères :
      - STAT : récupère le nombre et la taille des messages en attente
      - LIST ou UIDL : liste des messages à récupérer
      - RETR msg : permet de récupérer un msg
      - DELE msg : suppression
      - USER, PASS : login, passwd
      - QUIT : fin
        (ex : DELE 10, rep = OK, RETR 11, rep=OK+ msg, DELE 11,….)

IMAP

- Protocole permettant de relever le courrier sur un serveur
  - IMAP : Internet Message Access Protocol port 143, imaps 993
    - permet l'authentification si nécessaire de façon chiffrée
    - gère les mails sur le serveur, donc pas besoin de télécharger
    - permet de trier les mails, faire des répertoires, etc…
    - utilise des drapeaux pour la gestion des mails

    - IMAP est très utilisé pour les serveurs webmail.

  - Actuellement, version 4 rev 1, RFC 3501
    - Permet de gérer le mode on-line et aussi off-line
    - Utilisation de mots-clés : open, login, list, fetch, logout,…

Quelques fichiers

- Alias
  - permet de créer une liste de personne
    → liste de diffusion
  - permet de redéfinir le routage du courrier local
    → un utilisateur aura plusieurs noms possibles
    ex : laurenco@isima.fr, patrice.laurencot@isima.fr

- Le fichier .forward
  - fichier géré par l'utilisateur pour effectuer le re-routage des messages qui lui sont adressés
  - fichier consulté après les aliases

SPAM

- Courrier non sollicité envoyé à plusieurs personnes
- Actuellement , 95 % mail → SPAM !!
- Les adresses sont récupérées via les listes de diffusion, les pages Web, ou même achetés…

- **Solutions :**
  - Reconnaître l'auteur d'un SPAM
  - Filtrage au niveau personnel
  - Filtrage au niveau d'un site
    - liste noire des "spammeurs" connus
    - refuser les adresse invalides
    - interdire le relayage
    - refuser le mail pour qu'il soit renvoyé plus tard (efficace, mais cher en BP) -> liste grise