



Couche application

La **couche application** est la plus élevée du modèle de référence.

Elle est la **source et la destination** finale de toutes les **données** à transporter.



Couche application

La **couche application** gère les programmes de l'utilisateur pour lesquels les ordinateurs ont été achetés.

La couche fournit à ces utilisateurs quelques applications et services généraux que nous allons voir (**Telnet, FTP, ...**).



La gestion, l'accès et le transfert de fichier (1)

- Le transfert et l'accès à distance à un fichier sont parmi les applications les plus importantes d'un réseau.
- Les participants à un projet ont souvent besoin de partager des fichiers.
 - Une possibilité consiste à implanter les fichiers partagés sur une seule machine, et à organiser des transferts vers les autres systèmes lorsque le besoin s'en fait sentir.
 - L'autre consiste à laisser le fichier là où il a été créé, chaque utilisateur ayant la possibilité d'en prendre des copies, le choix se fait alors suivant la protection des répertoires.



La gestion, l'accès et le transfert de fichier (2)

- Les transferts d'information s'avèrent aussi nécessaires lorsque l'environnement des postes de travail sans disques est partagé à l'accès à une ou plusieurs stations regroupant les fonctions de stockage.
- Il est alors possible de travailler depuis n'importe quel station et de retrouver son environnement habituel (ex. Telnet, rlogin, ssh).



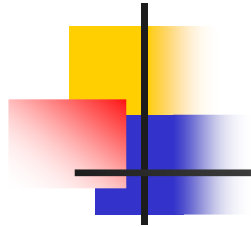
Le courrier électronique (1)

- L'avantage du **courrier électronique** vis à vis du téléphone est non seulement sa **rapidité**, son **faible coût** mais en plus il permet toujours de **joindre le correspondant** désiré, c'est à dire qu'il **ne nécessite** pas la **présence** des deux **interlocuteurs**, enfin il peut avoir **plusieurs destinataires**.
- C'est ainsi que le **courrier électronique** c'est très rapidement développé à **l'intérieur des réseaux**. Bien que le **courrier électronique** puisse être **considéré comme un simple cas particulier** du **transfert de fichier**, il **possède quelques caractéristiques bien spécifiques** :

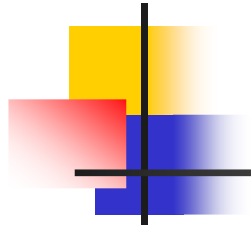


Le courrier électronique (2)

- En effet l'émetteur et le récepteur sont presque toujours des être humains et non des machines ce qui implique une interface utilisateur (composition, édition et lecture des textes).
- Aussi le systèmes électroniques traitent des documents fortement structurés c'est à dire qu'il comporte en général le nom, l'adresse du destinataire, la date et l'heure d'envoi, etc.....



Les protocoles de Haut niveau



Le protocole TELNET



Le protocole Telnet (1)

Telnet (**Telecommunication Network**) permet à une machine **client** de se **connecter** sur un **serveur**.

Une machine disposant d'un **serveur telnet** (ex. **telnetd** sous **Linux**) permettra à n'importe quelle machine de part le réseau de s'y connecter, au moyen d'un **client telnet**.

Les clients telnet existent sur la quasi-totalité des plates formes (**Windows, Linux, Unix, MacOS...**)



Le protocole Telnet (2)

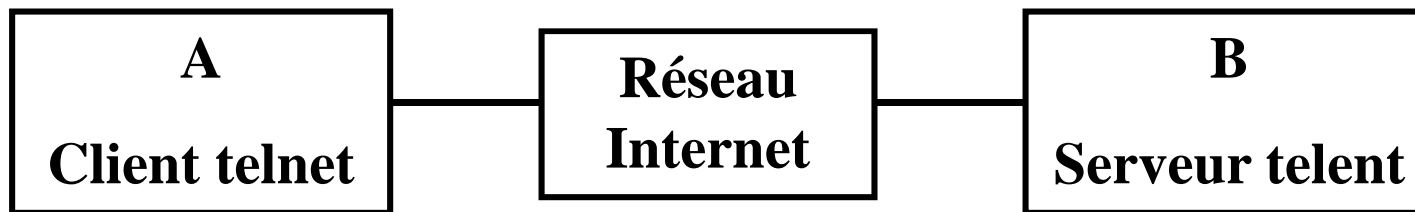
Lorsque une **machine A** désire se connecter à une **machine B**, A doit disposer d'un **client telnet** et B d'un **serveur telnet**.

Dans le monde Unix, le **serveur telnet** est **in.telnetd** et le client est **telnet**.

D'une **manière générale**, un serveur est souvent le **nom de service** proposé **suivi** de la lettre '**d**': **Exemple: telnetd**



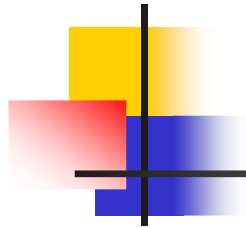
Connexion telnet de A vers B(1)



Lorsque A envoie une requête à la machine B, celle-ci est à l'écoute permanente des requêtes présentées sur le port TCP (23 par défaut).

B répond alors par une demande d'authentification, à laquelle A doit répondre (login + passwd).

Lorsque cette phase est réussie, l'entrée standard est redirigée sur le clavier de A, et la sortie standard est redirigé sur l'écran de A.



Connexion telnet de A vers B (2)

Tout se passe comme si l'utilisateur de **A** était devant la machine **B** alors que des milliers de kilomètres peuvent les séparer.

Le **protocole Telnet** est basé sur le protocole **TCP**. C'est donc la couche **TCP (TCP/IP)** qui se **charge d'assurer** l'intégrité de **l'échange des données**.

Telnet est un protocole de haut niveau qui **ignore** tout de **l'architecture** utilisé, soit sur la **machine** soit sur la **topologie** du réseau.

Il ne fait que renvoyer un **echo** des **données envoyées** ou **reçues**.



Exemples de connexion telnet

```
[desa@desa ~/desaIT/DESA1 (4)]$ telnet koutoubia.fsr.ac.ma
```

```
Trying 192.16.1.20.....
```

```
Connect to koutoubia.fsr.ac.ma
```

```
Espace character is '^']'.
```

```
redHat Linux release 4.2
```

```
Kernel 2.0.30 an i586
```

```
Login : DESA
```

```
Passwd: ****
```

```
Last login : sun Sep 28 20:27:40 on tty1
```

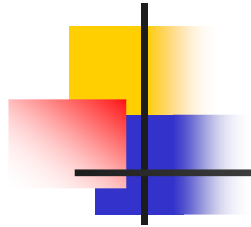
```
[dESA@DESA~/ (1)]$ whoami
```

```
DESA
```

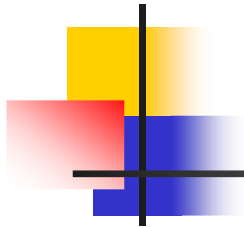
```
[dESA@DESA~/ (2)]$ logout
```

```
Connection closed foreign host
```

```
[desa@desa ~/desaIT/DESA1 (5)]$
```



Le protocole FTP



Le protocole FTP (1)

FTP : **File Transfer Protocol**, est utilisé dès qu'il s'agit de transférer des données entre **deux machines A et B**.

Comme en **Telnet**, la machine **A** doit être équipé d'un **client FTP**, alors que la **machine B** doit elle équipée d'un **serveur FTP**.

Lorsque **A envoie** une **demande** de **connexion** à **B**, le **serveur FTP** **renvoie** alors le **message de login** défini par l'administrateur de **B**.

L'utilisateur de **A** envoie alors la commande **USERlogin** attendue par **B**, où **login** est le **nom de l'utilisateur**.

B attend alors la **commande Password**, où password est le **mot de passe** correspondant à l'utilisateur **login**.



Le protocole FTP (2)

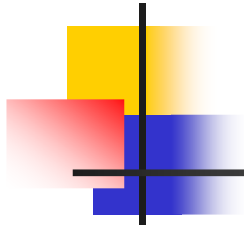
De nombreux serveurs autorisent les connexions dites **anonymes**, c'est-à-dire que **n'importe quel utilisateur peut s'y connecter** pour prendre des fichiers, ou sur certains serveurs **déposer des fichiers**.

Dans le **phase de connexion anonyme**, on envoie généralement **FTP** ou **anonymous** comme **nom de login** et **son adresse email en mot de passe**.

Une fois **l'identification effectuée**, le **client envoie la commande SYST**, de manière à **connaître le système distant**.

A cette étape, vous avez **ouvert un canal de commande** sur le **port TCP 21** mais vous **n'êtes pas prêts à transférer des données**.

Les **données** sont transférées sur le **port TCP 20**.



Le protocole FTP (3)

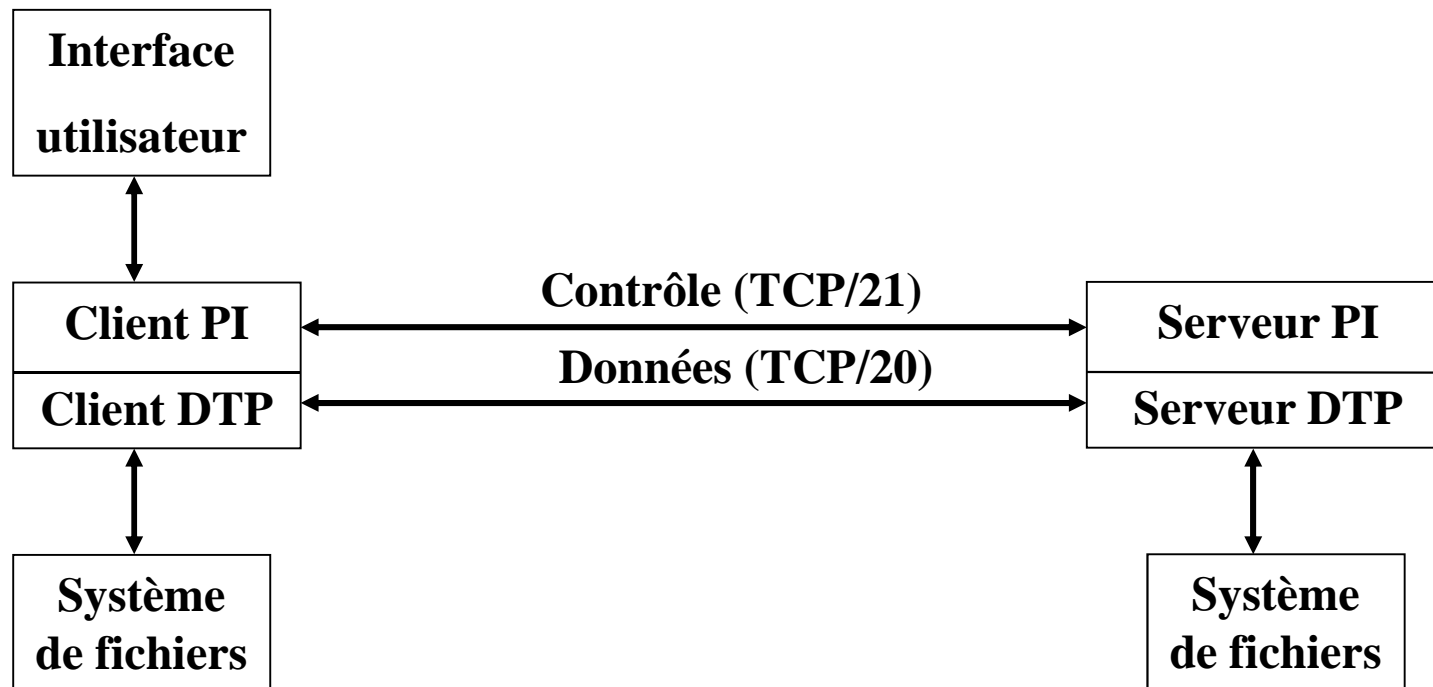
Le protocole **FTP** utilise par convention le **port TCP/21** pour les **commandes** et le **port TCP/20** pour les **données**.

Le port TCP/21 est appelé l'interpreteur de protocole (**Protocol Interpreter** ou **PI**)

Le port TCP/20 est appelé processus de transfert de données (**Data Transfer Process** ou **DTP**).

Une fois le **canal de données est ouvert**, vous pourrez **transférer** des **données** entre les **machines A et B**.

Connexion FTP de A vers B (1)





Connexion FTP de A vers B (2)

Remarque:

A l'heure actuelle, le **protocole FTP** ne **conserve pas les droits d'accès** sur les **fichiers transférés**.

FTP utilise **deux modes** de transfert, **ASCII** ou **Binaire**. Pensez à vous placer dans le bon mode avant d'entamer un transfert.

FTP effectue tous ses transferts en **avant-plan**, donc vous devez attendre que les transferts soient achevés avant d'entamer un autre transfert.



Exemples de connexion FTP

```
[desa@desa ~/desaIT/DESA1 (7)]$ ftp localhost
```

```
Connected to localhost
```

```
220 desa.image.fsr.ac.ma ftp server (version wu-2.4.2-academ [Beta-14](1)) Thu Sep 11 00:49:43  
MET DST 1997) ready.
```

```
Name (localhost:desa): ftp ou anonymous
```

```
331 Guest login ok, send yur complet e-mail adress as password.
```

```
Password:*****
```

```
230 Guest login ok, acces resrctions apply.
```

```
Remote system type UNIX.
```

```
Using binary mode to transfert files.
```

```
ftp> ls
```

```
200 PORT comand successful.
```

```
150 Opening ASCII mode data connection for /bon/ls
```

```
Total 20068
```

```
drwxr-xr-x 7 root root 1024 sep 25 19:40
```

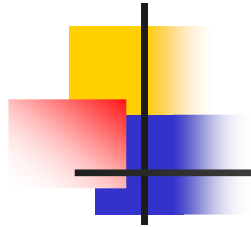
```
-rw-r--r-- 1 root root 42110 sep 10 21:47 vmlinuz
```

```
226 transfer complete.
```

```
ftp> bye
```

```
221 Goodbye
```

```
[desa@desa ~/desaIT/DESA1 (8)]$
```



Le protocole SMTP



Le protocole SMTP (1)

SMTP: Simple Mail Transfert Protocol

Protocole qui permet le **transfert des courriers électroniques**.

Il est **similaire** au protocole **FTP**, de **part son langage de commande**.

Il est implémenté sur le port **TCP/25**.

Sur **UNIX**, **sendmail** est utilisé comme **client** et comme **serveur**.



Le protocole SMTP (2)

SMTP utilise des **files d'attente** pour gérer les transferts de courrier.

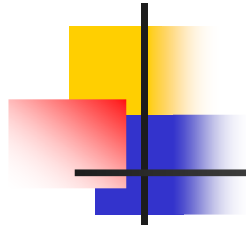
Lorsqu'un **message** est **envoyé** au **serveur SMTP**, celui-ci le **place** dans une **file d'attente**, puis **tente de le livrer** à la machine de destination.

Si la machine **destination** n'est pas accessible, le **serveur SMTP** tentera selon **la configuration** de le transmettre ultérieurement.

Tous les messages sont transférés dans un **format ASCII**.

La **fin d'un message** est indiquée par un **‘.’** sur une **ligne vierge**.

Le protocole SMTP (3)



Echange de courrier entre deux serveurs SMTP

Dans la phase d'échange de courrier entre deux serveur SMTP, la première phase est l'authentification de la machine émettrice.

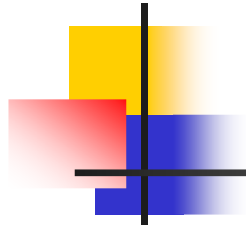
La machine qui demande la connexion envoie: la commande HELO suivi de son nom de domaine.

La machine réceptrice renvoie un message de bienvenue et présente les commandes disponibles.

La machine émettrice va maintenant donner le nom de l'expéditeur, par la commande MAIL FROM: login.

Ensuite, l'émetteur indique à qui s'adresse ce courrier, par la commande RCPT TO: login.

Le protocole SMTP (4)



Echange de courrier entre deux serveurs SMTP

A ce niveau, les machines sont prêtes à échanger les messages.

La machine émettrice envoie alors la commande DATA, puis termine cette phase de transfert du message en envoyant ‘.’

La connexion reste alors établie, et les deux machines peuvent continuer à transférer des courriers, ou retourner leur mode de connexion (émettrice → réceptrice et inversement).

Si plusieurs destinataires sont spécifiées dans le champ RCPT le message est alors envoyé à tous les destinataires.

Exemple de dialogue SMTP (1)



```
[desa@desa ~/desaIT/DESA1 (7)]$ telnet localhost 25
```

```
Trying 127.0.0.1
```

```
Connected to localhost
```

```
Escape character is '^'
```

```
220 desa.image.ac.ma SMTP Sendmail 8.8.4/8.8.4; Thu Sep 11 00:49:43 1997.
```

```
Helo image.fsr.ac.ma
```

```
250 desa.image.ma.ac hello localhost [127.0.0.1], pleased to meet you
```

```
250-EXPN
```

```
250-SIZE
```

```
250-HELP
```

```
Mail from: desa@desa.image.ac.ma
```

```
250 desa@desa.image.ac.ma ...Sender ok
```

```
Rcp to: desa@desa.image.ac.ma ... Recipeint Ok
```

```
Data
```

```
354 Enter mail, end with «. » on line by itself
```

```
Un exemple de message sans client, directement, sur le port SMTP 25/TCP
```

```
.
```

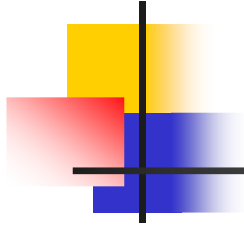
```
250 UAA00614 message accepted for delivery
```

```
Quit
```

```
221 desa.image.ac.ma closing connection
```

```
[desa@desa ~/desaIT/DESA1 (8)]$
```

Dialogue SMTP (2)



Lorsque le message a été correctement acheminé sur la machine de destination celui-ci est ensuite placé en attente dans la boîte aux lettres de l'utilisateur.

Sur la plupart des configurations UNIX, les messages sont placés dans le répertoire `/var/spool/mail/`.

Chaque **utilisateur** a un **fichier** sous **son nom de login**, qui **contient** les **messages**.

Chaque message **possède une entête qui renseigne la provenance du message, l'heure à laquelle il est arrivé, etc.....**



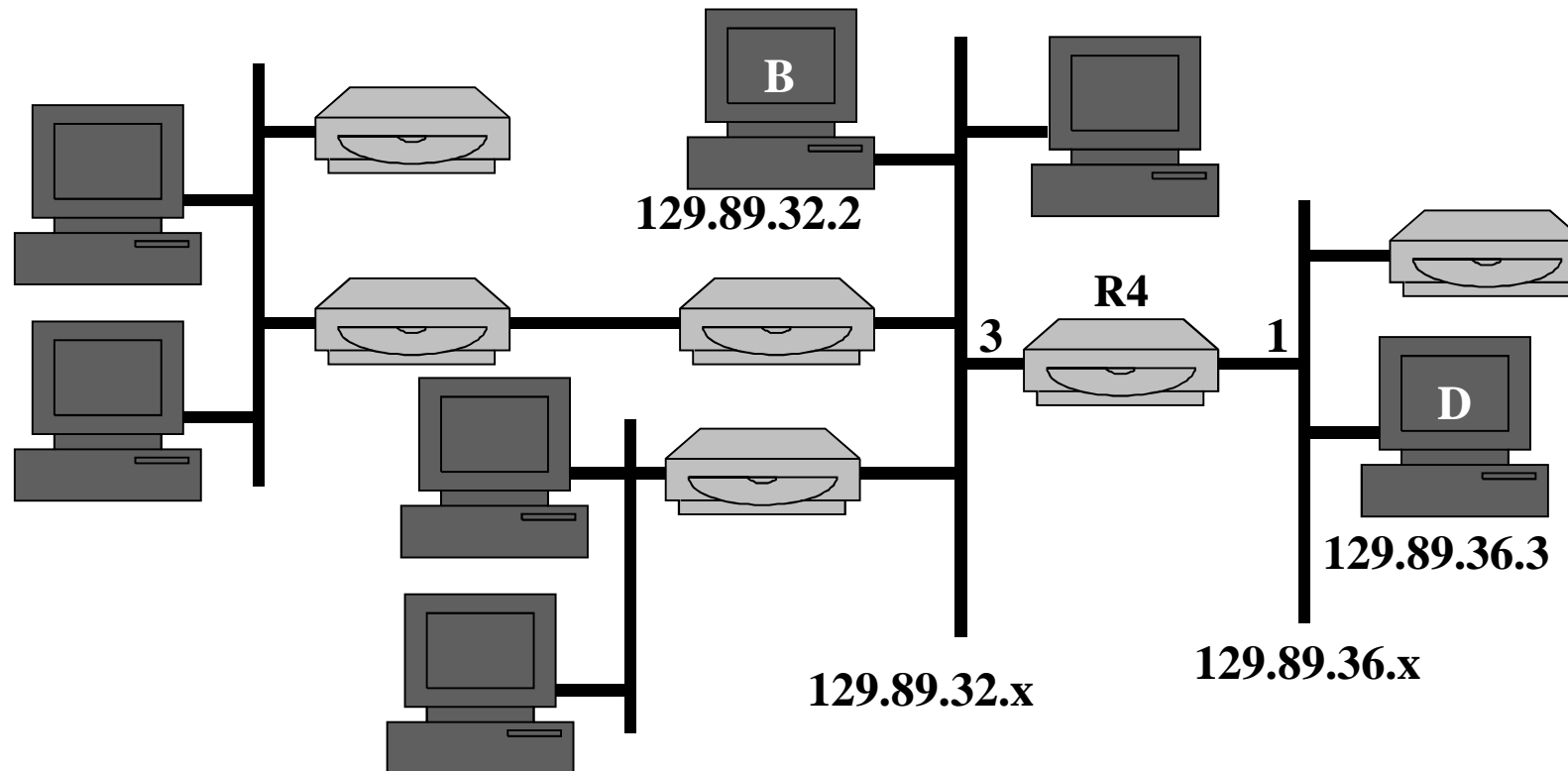
Exemple d'un message dans une boite aux lettres

From: desa@desa.image.ac.ma Thu Sep 11 00:51:03 1997
Return Path: <desa@desa.image.ac.ma>
Received: from desa.image.ac.ma (localhost [127.0.0.1])
By desa.image.ac.ma with SMTP
Data: Thu Sep 11 00:49:43 1997
From : DESA IT2 <desa@desa.image.ac.ma>
Message-ID [199709111841.UAA00614@desa.image.ac.ma](#)

Un exemple de message sans client, directement, sur le port
SMTP 25/TCP

Exemple de connexion Telnet (1)

Que se passe-t-il, lorsque un utilisateur sur la **machine B**, entre la command: **Telnet machine D**?





Exemple de connexion Telnet (2)

La machine B (adresse 129.89.32.2) traduit le nom « machine D » en adresse IP 129.89.36.3, d'après sa table hosts (liste des correspondances nom de la machine-adresse IP)

ou d'après interrogation d'un serveur de nom qui fait la correspondance entre le nom de la machine et l'adresse IP.

Si la machine B ne trouve pas l'adresse correspondante, le message unknown host s'affiche.



Exemple de connexion Telnet (3)

Comment atteindre 129.89.36.3 ?

Ce n'est pas un numéro 129.89.32.x, il faut donc passer par un routeur.

B regarde sa table de routage pour connaître par quel routeur elle doit passer, il faut passer par le routeur R4 (d'adresse IP 129.89.32.3).

B émet une trame «broadcast » Ethernet après avoir vérifié qu'elle n'a pas déjà l'information dans sa table ARP (qui donne la correspondance adresse IP/ adresse Ethernet) .

Cette trame demande qu'elle est l'adresse Ethernet de 129.89.36.3.



Exemple de connexion Telnet (4)

Le routeur R4 répond à B (RARP): l'adresse Ethernet de 129.88.36.3 est 00:00:c0:00:5b:37

B envoie une trame Ethernet avec l'adresse de destination 00:00:c0:00:5b:37, incluant un datagramme IP (adresse origine 129.89.32.2 et adresse destination 129.89.36.3) contenant un segment TCP avec un numéro de port destinataire pour Telnet (port 23).

R4 reçoit la trame Ethernet, extrait le datagramme IP, l'adresse IP du destinataire et cherche où l'envoyer.

Il a une interface (129.89.36.1) de la forme 129.89.36.x. Il sait donc que 129.89.36.3 est sur le segment Ethernet de son interface 129.89.36.1.



Exemple de connexion Telnet (5)

R4 recherche alors l'adresse Ethernet de 129.89.36.3. S'il ne le trouve pas dans sa table ARP, il envoie un broadcast ARP. Il peut ensuite envoyer le datagramme IP à la machine D.

D reçoit le datagramme IP, il extrait le segment TCP, ouvre une session TCP, avec l'indication du port numéro 23, il appelle le service associé ici TELNET.

Telnet demande le nom de l'utilisateur, la question est transportée par un segment TCP, dans un datagramme IP (@ origine 129.89.36.3 et @ destination 129.89.32.2)

Pour envoyer ce datagramme, la machine D cherche l'itinéraire avec la même méthode que la machine B au départ.



Exemple de connexion Telnet (6)

Remarques:

D ne tient pas compte de la précédente arrivée d'un datagramme IP pour trouver l'itinéraire de la réponse. Il refait le raisonnement, comme s'il n'avait rien reçu.

Pour tester la connectivité IP, il n'est pas utile de tester un appel de B vers A, si on a déjà testé un appel de A vers B.

Le broadcast ARP n'est pas utilisé que lors de la première recherche d'adresse Ethernet.