

# Architecture des réseaux et des systèmes informatiques répartis

Théo Pierron

Polytech 4A Apprentissage

FTP et TFTP

FTP

# File Transfer Protocol

- Transfert de fichiers
- Indépendant de l'OS
- Fournit une interface UNIX-like
- Sur le modèle client/serveur basé sur TCP

## Authentification par mot de passe

- Gestion des permissions
- Cantonnement d'un utilisateur à un sous-dossier

## Authentification par mot de passe

- Gestion des permissions
- Cantonnement d'un utilisateur à un sous-dossier

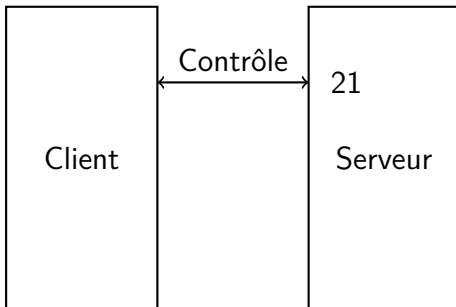
## Mode anonyme

- Login = anonymous
- Mot de passe = ce qu'on veut (généralement une adresse mail)

# Démo

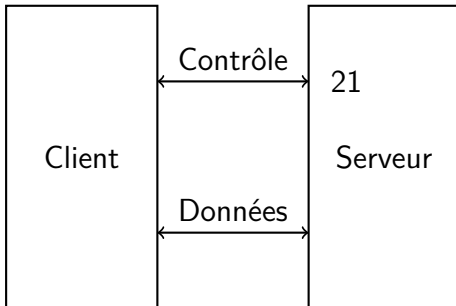
# Fonctionnement

Deux sockets : contrôle + données



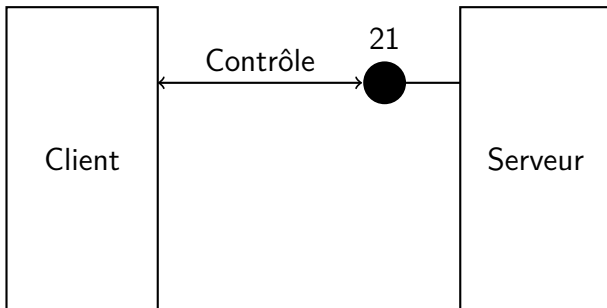
# Fonctionnement

Deux sockets : contrôle + données

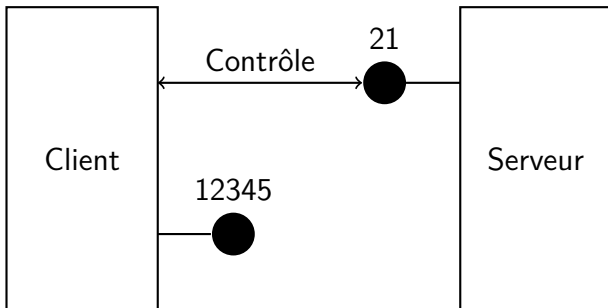




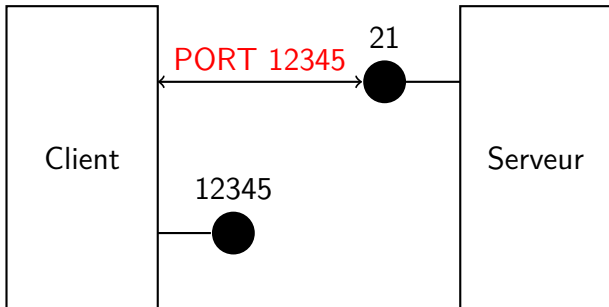
# Mode actif



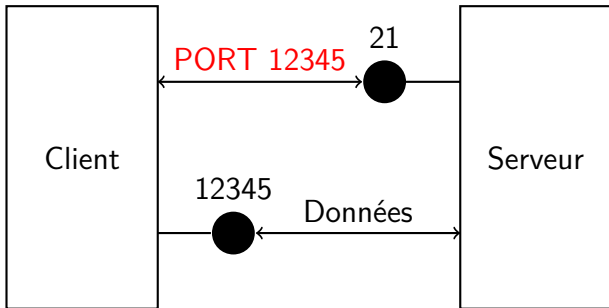
# Mode actif



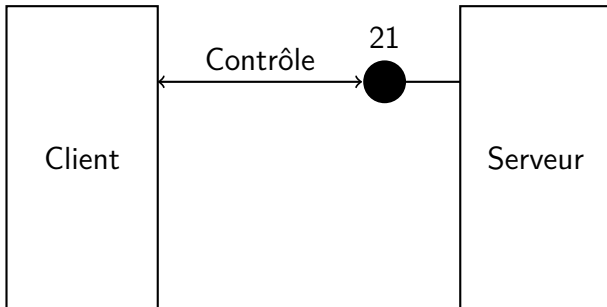
# Mode actif



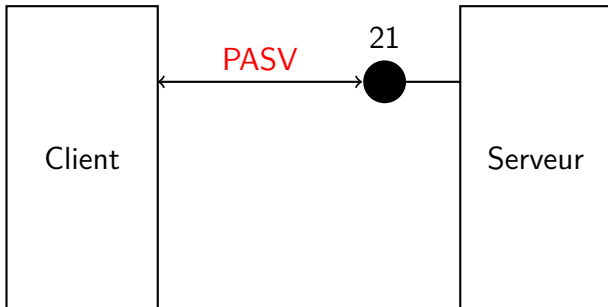
# Mode actif



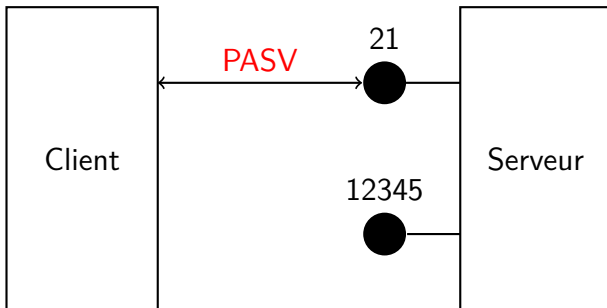
# Mode passif



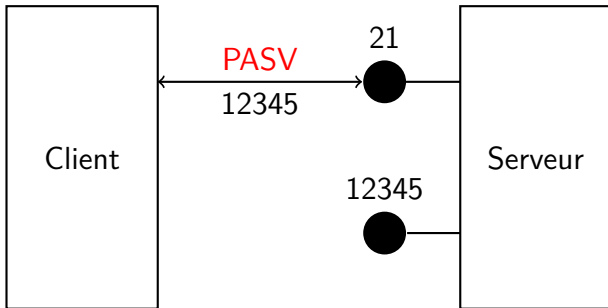
# Mode passif



# Mode passif

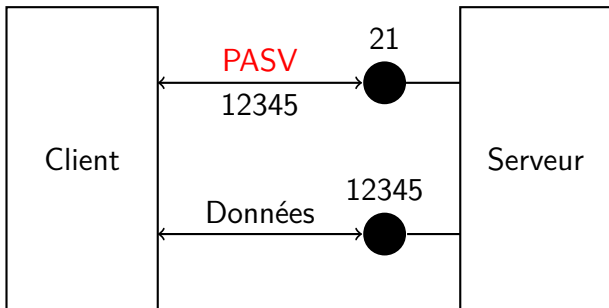


# Mode passif





# Mode passif



Pas de sécurité :

- Échanges en clair
- Utiliser SFTP (ssh), FTPS (SSL),...

# TFTP

# Trivial File Transfer Protocol

- Protocole minimaliste
- Optimisation de la vitesse
- Basé sur UDP
- Nécessite un réseau fiable
- Boot à distance, mise à jour d'équipements réseau

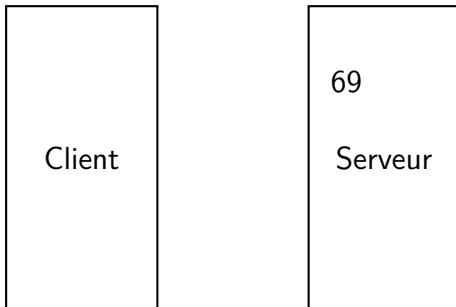
# Fonctionnalités

- Lire un fichier (nécessite de connaître le nom à l'avance)
- Écrire un fichier
- Pas de 1s, pas d'authentification

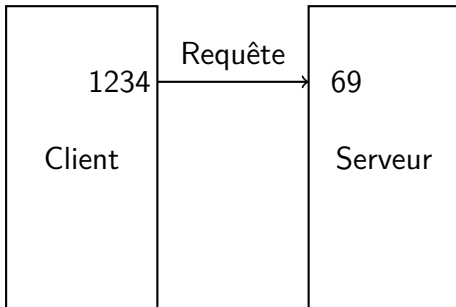
# Le protocole

- Le serveur attend une requête sur le port 69.
- Identification du client par le port d'émission.
- Transfert par bloc de 512 octets. Bloc plus petit  $\Rightarrow$  fin du transfert.
- Chaque packet doit être acquitté.
- Si ACK pas reçu après un certain temps, on réémet le dernier paquet émis.

# Initialisation

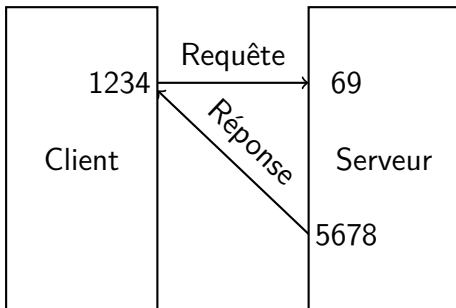


# Initialisation





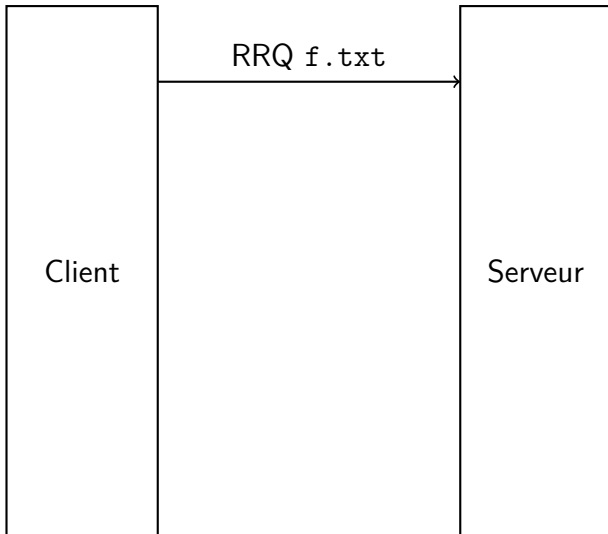
# Initialisation



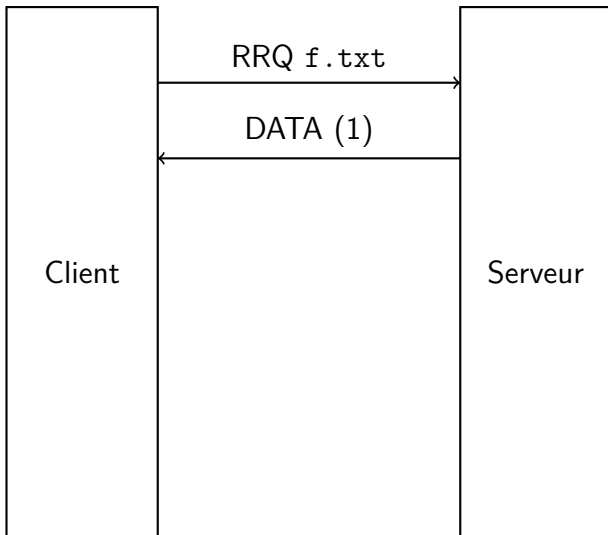
## 5 types de datagrammes

- ReadReQuest
- WriteReQuest
- ACK
- DATA
- ERROR

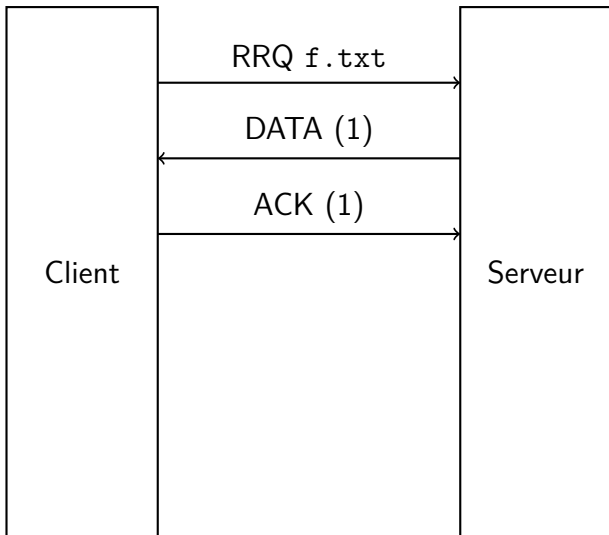
## Lecture de 1000o



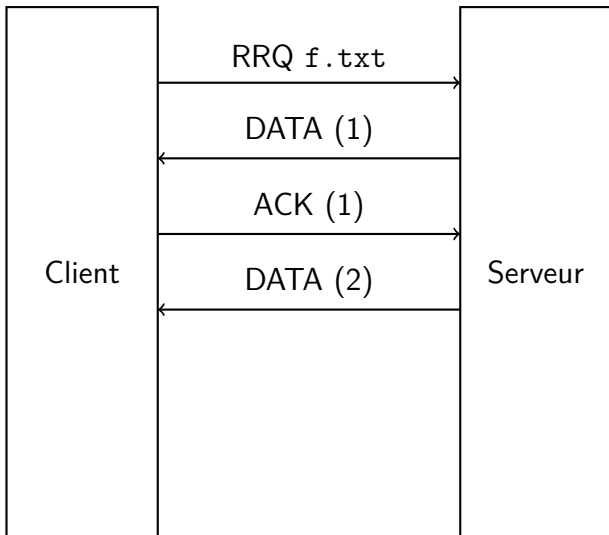
## Lecture de 1000o



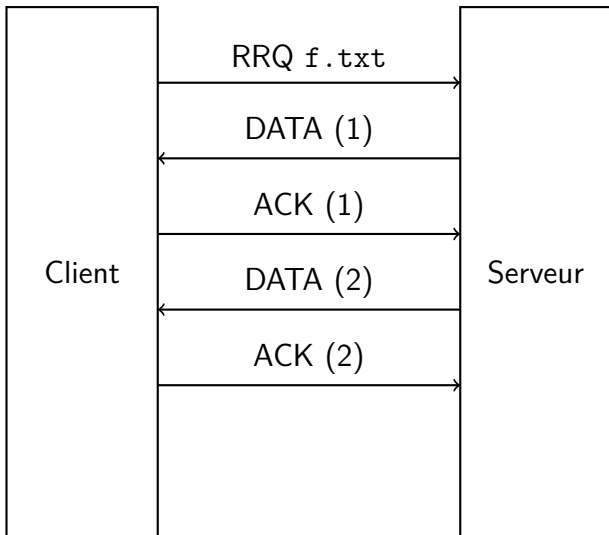
## Lecture de 1000o



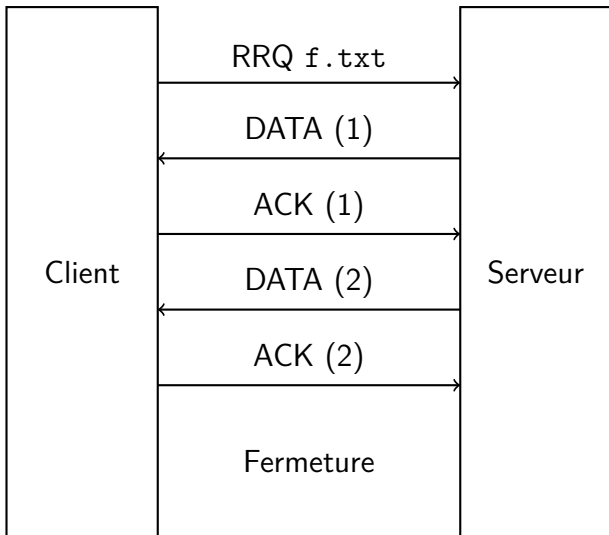
## Lecture de 1000o



## Lecture de 1000o

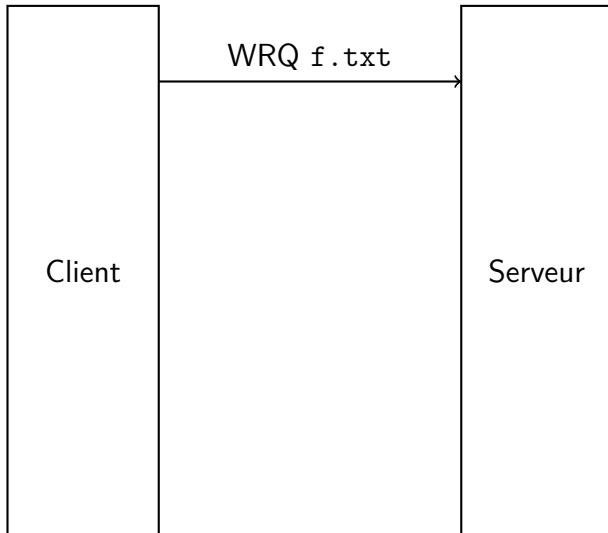


## Lecture de 1000o

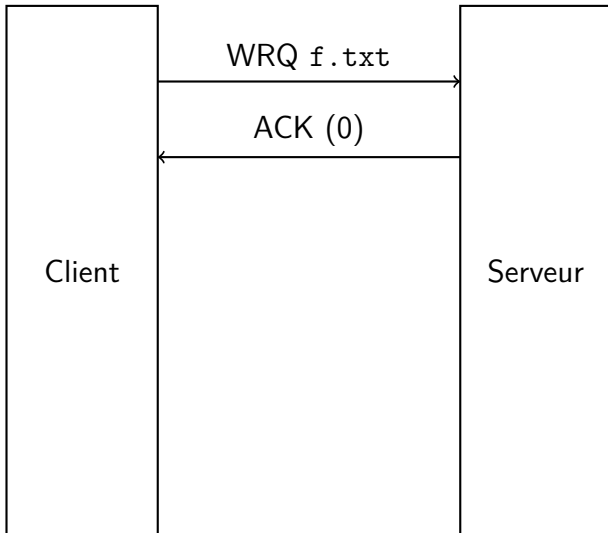




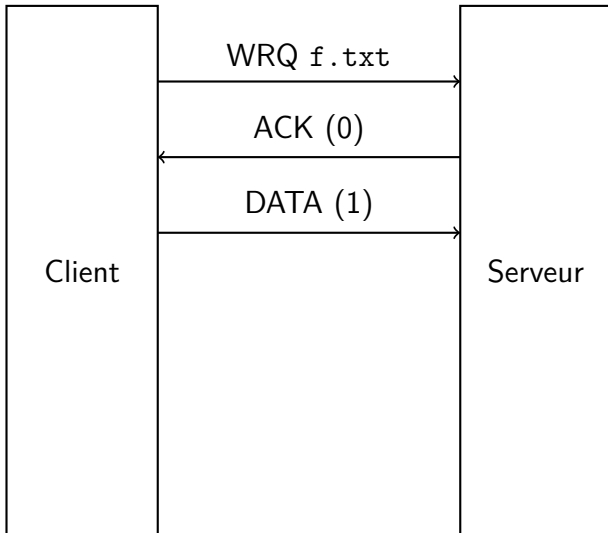
## Écriture de 1000o



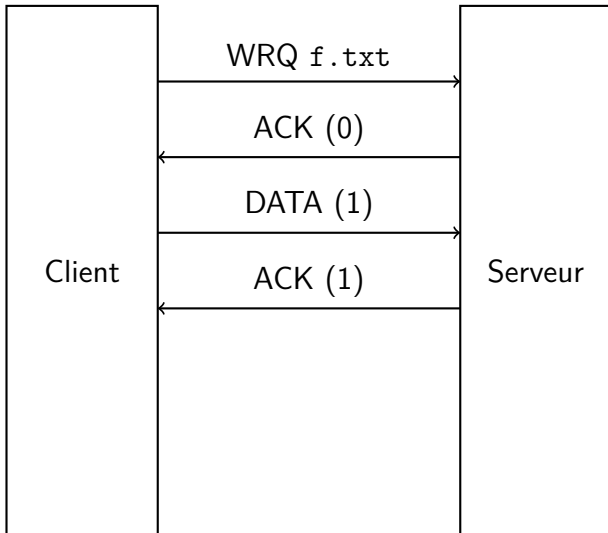
## Écriture de 1000o



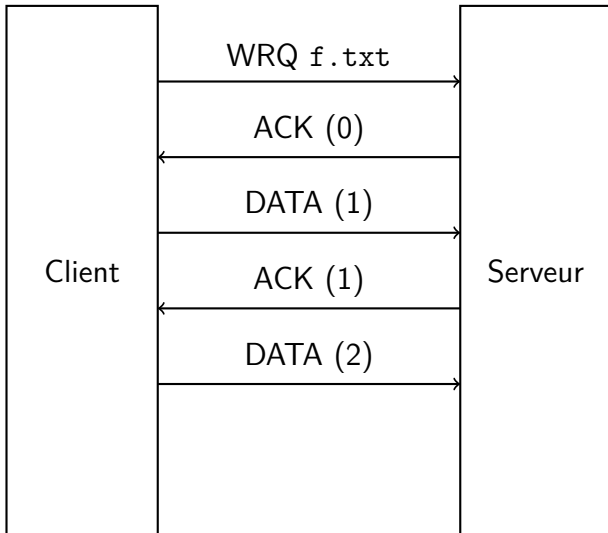
## Écriture de 1000o



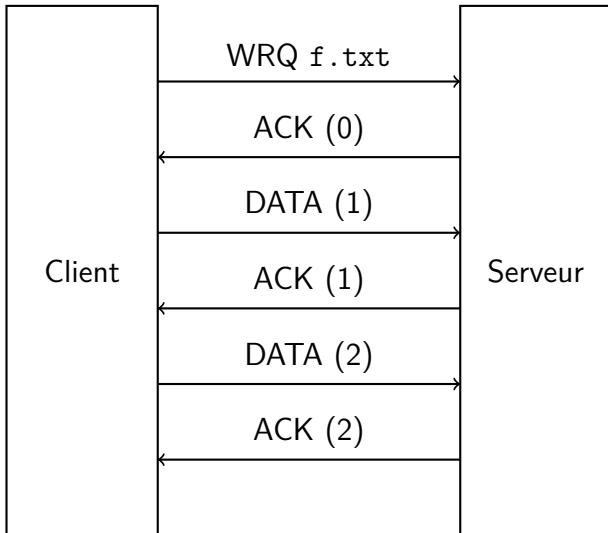
# Écriture de 1000o



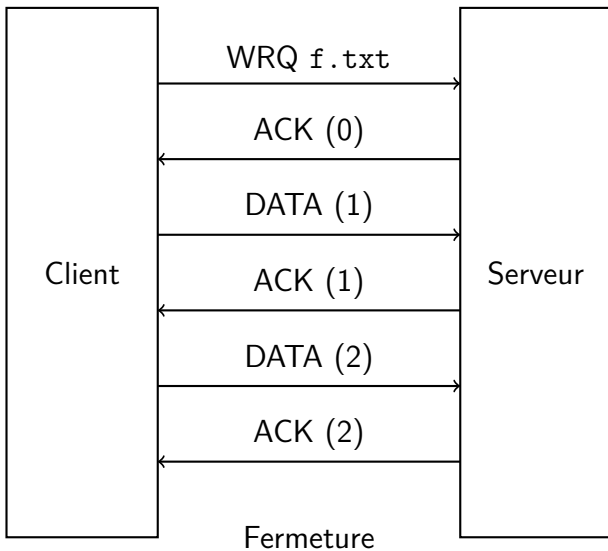
# Écriture de 1000o



# Écriture de 1000o



# Écriture de 1000o



## RRQ/WRQ

- 2 bytes : op code (1 pour read, 2 pour write)
- le nom du fichier
- 1 byte nul
- Mode : netascii/octet/mail
- 1 byte nul



# DATA/ACK

- 2 bytes : op code (ici, 3)
- 2 bytes : numéro de bloc
- 512 bytes : données

## DATA/ACK

- 2 bytes : op code (ici, 3)
- 2 bytes : numéro de bloc
- 512 bytes : données
- 2 bytes : op code (ici, 4)
- 2 bytes : numéro de bloc

# ERROR

Contenu :

- 2 bytes : op code (ici, 5)
- 2 bytes : code d'erreur (7 cas)
- message d'erreur
- 1 byte nul

# ERROR

## Contenu :

- 2 bytes : op code (ici, 5)
- 2 bytes : code d'erreur (7 cas)
- message d'erreur
- 1 byte nul

## Remarques :

- Un paquet ERROR vaut acquittement, et n'est pas acquitté.
- La réception d'un paquet ERROR ferme la connexion, sauf s'il n'a pas le bon port source.
- Dans ce cas, on répond un ERROR à l'émetteur.