

Protocoles « Application »

- Nombreuses applications TCP/IP:
 - POP3
 - TFTP
 - BOOTP & DHCP
 - service de configuration automatique
 - FTP
 - Etc.

Application POP3

Documentation

Post Office Protocol 3, RFC 1460
(Request For Comments)

But

Protocole TCP pour la réception
des messages d'une boîte à
lettre distante.

Protocole Application POP3

■ Principes

Dialogue client-serveur sur port réservé POP3 (port 110).

Chaque requête du client a pour réponse une ligne commençant par +OK ou -ERR. Les lignes sont terminées par CR LF.

■ Connexion au serveur

Réponse du serveur:

```
+OK Qpopper (version 4.0.4) at stargate  
starting.
```

Protocole Application POP3

- **S'identifier**

USER felixp

+OK Password required for felixp.

PASS Devine!!!

+OK felixp has 3 visible messages (0 hidden) in 4235 octets.

- **Voir le nombre et la liste des messages**

STAT

+OK 3 4235

LIST

+OK 3 visible messages (4235 octets)

1 1508

2 1464

3 1263

.

■ Lire un message

RETR 3

+OK 1263 octets

Return-Path: <felixp@ist.ga>

Received: from komet ([172.16.94.87])

*by stargate.ist.ga (8.12.3/8.12.3) with SMTP id g4S9hEpN004334
for <felixp@ist.ga>; Tue, 28 May 2002 10:43:20 +0100*

Message-ID: <008201c2062b\$4c266420\$575e10ac@ist.ga>

Reply-To: "Patrick FELIX" <felix@labri.fr>

From: "Patrick FELIX" <felixp@ist.ga>

To: <felixp@ist.ga>

Subject: un message

Date: Tue, 28 May 2002 10:37:29 +0100

.../...

X-Mailer: Microsoft Outlook Express 6.00.2600.0000

Un exemple de message .../...

.

REMARQUE : les lignes de texte qui commencent par un point sont précédées par un autre point, pour les distinguer de la fin du message (byte-stuffing).

- **Détruire un message**

DELE 2

+OK message 2 marked for deletion

- **Annuler les destructions**

RSET

+OK 3 messages ready

- **Quitter**

+OK POP3 Server shutdown.

Application TFTP

Documentation

Trivial File Transfert Protocol,
RFC 783 et 1350

But

Protocole UDP pour le transfert
'simple' de fichiers sans
identification.

Principes

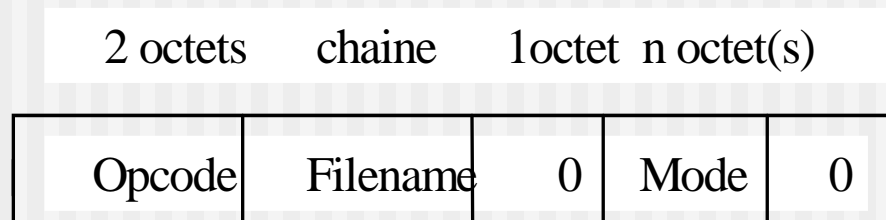
- « Connexion »
 - demande de lecture ou d'écriture
- 5 types de paquet:
 - RRQ, WRQ, DATA, ERR, ACK
- transfert par bloc de 512 octets
- acquittement
- retransmission si perte

Phase de connexion

- TID (Transfert IDentification)
- le client choisit un TIDsource et prend TIDdest=69 (port réservé tftp)
- la réponse positive du serveur contient le TIDsource à utiliser ensuite

Phase de connexion

- requête de lecture(RRQ) ou écriture(WRQ)

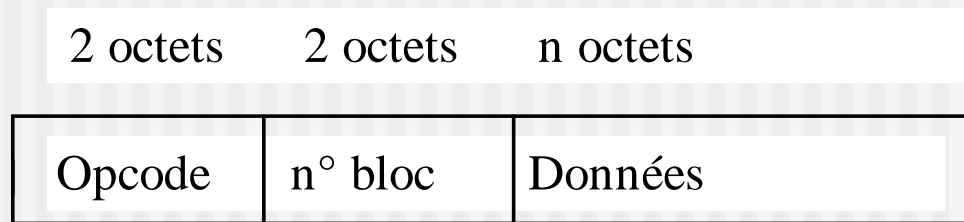


Opcode RRQ = 1
WRQ = 2

- réponse : ACK(acquittement=0 pour WRQ) ou DATA(pour RRQ), ou ERROR

Transfert de données

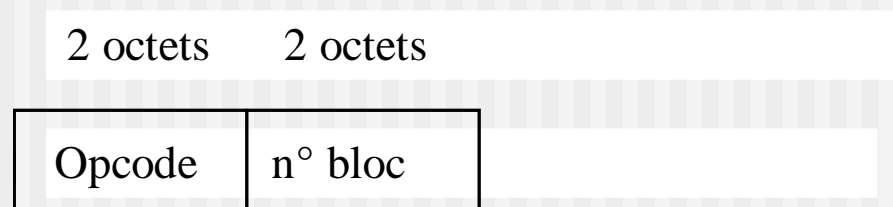
- paquet DATA



- opcode = 3
- numéro de bloc: commence à 1
- taille maximale des données: 512 octets

Transfert de données

- paquet ACK



- Ack Opcode = 4
- WRQ et DATA acquittés par ACK ou ERR
- RRQ acquitté par DATA ou ERR

Traitement des erreurs

- paquet ERROR
 - Opcode=5
 - code erreur:
 - (0) non défini, voir message.
 - (1) fichier non trouvé
 - (2) violation d'accès
 - ...

Terminaison

- terminaison normale
 - paquet DATA < 512 octets
 - dernier paquet DATA acquitté

- terminaison prématurée
 - paquet ERROR

Exercices

- Pourquoi des timers doivent être utilisés ?
- Expliquez pourquoi des acquittements peuvent être dupliqués ?
- Donner un scénario de réception, par un client TFTP, d'un fichier de 1500 octets dans chacun des cas suivants :
 - Transmission sans erreur,
 - Erreur lors de la transmission du 2^{ième} bloc.
 - Serveur TFTP en panne.
- Analyser la trace Ethernet suivante :

Trace Ethernet (tftp)

UDP from atlantide.50559 to firmin.tftp 27 bytes

08 00 20 10 b6 7c 08 00 20 75 13 88 08 00 45 00 00 2f 3c 22 40 00 ff 11 07 55 93

*d2 08 03 93 d2 08 9f c5 7f 00 45 00 1b a0 d2 00 01 6c 6b 6a 66 64 73 66 00
6e 65 74 61 73 63 69 69 00*

UDP from firmin.3940 to atlantide.50559 27 bytes

08 00 20 75 13 88 08 00 20 10 b6 7c 08 00 45 00 00 2f 25 30 00 00 3c 11 21 48 93

*d2 08 9f 93 d2 08 03 0f 64 c5 7f 00 1b 00 00 00 05 00 01 46 69 6c 65 20 6e 6f 74
20 66 6f 75 6e 64 00*

UDP from atlantide.50559 to firmin.3940 12 bytes

08 00 20 10 b6 7c 08 00 20 75 13 88 08 00 45 00 00 20 3c 23 40 00 ff 11 07

63 93 d2 08 03 93 d2 08 9f c5 7f 0f 64 00 0c f2 a6 00 04 00 01 00 00 00 20

00 00 00 01 ff ff ff ff ff ff

Trace tftp

UDP from atlantide.50559 to firmin.tftp 27 bytes

08 00 20 10 b6 7c 08 00 20 75 13 88 08 00 45 00 00 2f 3c 22 40 00 ff 11 07 55 93
d2 08 03 93 d2 08 9f c5 7f 00 45 00 1b a0 d2 00 01 6c 6b 6a 66 64 73 66 00 6e 65
74 61 73 63 69 69 00

UDP from firmin.3940 to atlantide.50559 27 bytes

08 00 20 75 13 88 08 00 20 10 b6 7c 08 00 45 00 00 2f 25 30 00 00 3c 11 21 48 93
d2 08 9f 93 d2 08 03 0f 64 c5 7f 00 1b 00 00 00 05 00 01 46 69 6c 65 20 6e 6f 74
20 66 6f 75 6e 64 00

Trace tftp

UDP from atlantide.50559 to firmin.3940 12 bytes

08 00 20 10 b6 7c 08 00 20 75 13 88 08 00 45 00 00 20 3c 23 40 00 ff 11 07
63 93 d2 08 03 93 d2 08 9f c5 7f 0f 64 00 0c f2 a6 **00 04 00 01 00 00 00 20**
00 00 00 01 ff ff ff ff ff ff

Applications BOOTP & DHCP

Résolution d'adresses et configuration automatique

- ARP (IP \Rightarrow physique)
- RARP (physique \Rightarrow IP)
RARP/ARP: bas niveau

- BOOTP, DHCP
 - s'appuient sur UDP / IP,
 - véhiculent l'ensemble des paramètres 'réseau'

Au démarrage d'une station...

Récupération de sa configuration réseau :

- Soit RARP + ICMP (netmask) + TFTP un peu lourd... et pas optimal
- Soit BOOTP / DHCP (+ TFTP si sans disque uniquement)

Application BOOTP

Documentation

BOOTstrap Protocol, RFC
951 et 1542

But

Protocole UDP, pour fournir
des informations de
configuration.

Principe

- Serveur : port 67
- Client : port 68
- Le client envoie en broadcast un paquet BOOTREQUEST contenant au minimum son adresse physique
- Le serveur répond avec un paquet BOOTREPLY contenant tous les paramètres utiles (IP du client, netmask, IP serveurs, passerelles, etc...)

Application DHCP

Documentation

Dynamic Host Configuration
Protocol

RFC 2131 et RFC 2132

But

Extension de BOOTP: plus de
paramètres pris en compte, et
allocation automatique
d'adresses IP.

Application FTP

Documentation

File Transfert Protocol,
RFC 959

But

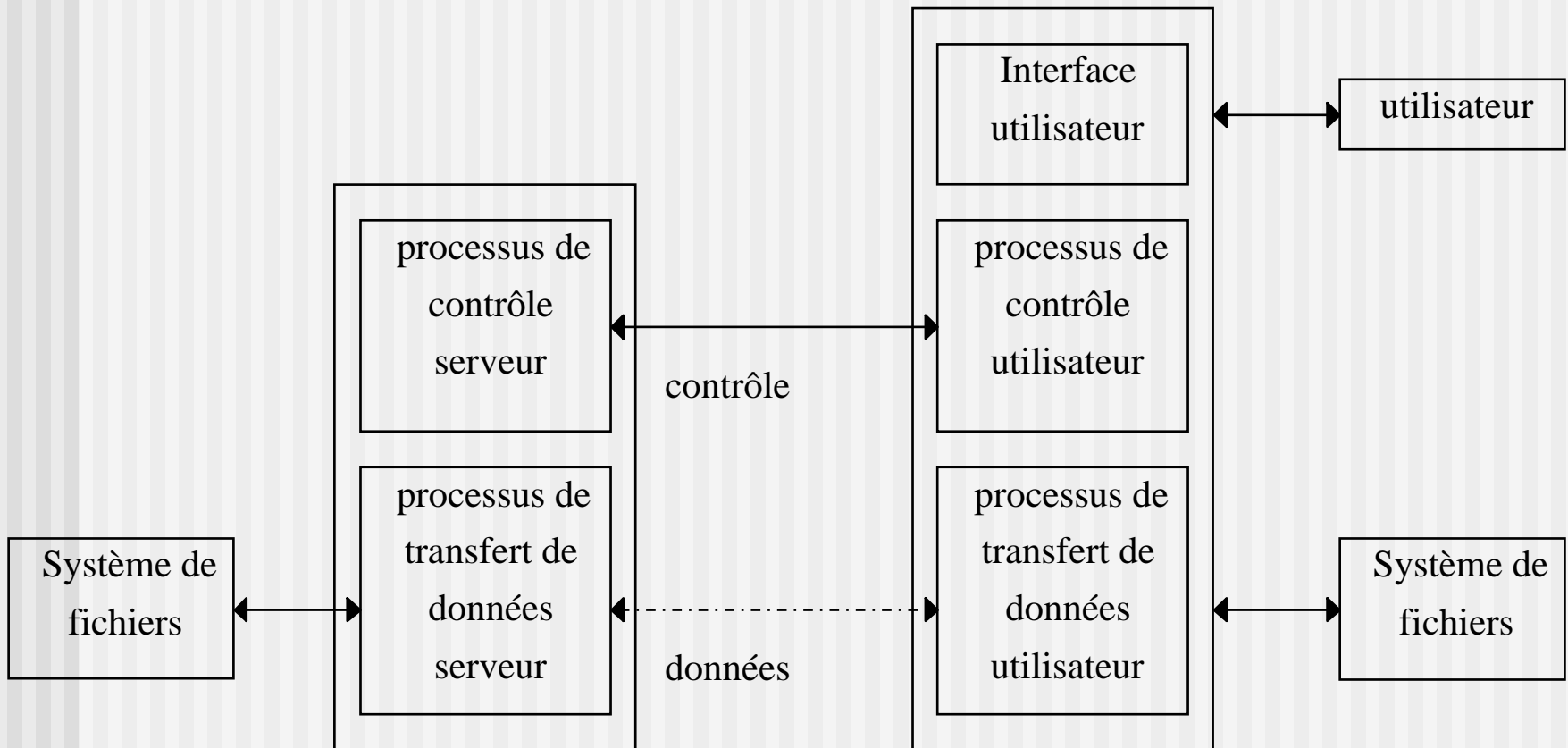
Transfert de fichiers
avec identification

FTP : File Transfert Protocol

■ Principes

- Accès interactif
- Spécification de la représentation
- Vérification de l'identité du client
- Utilise TCP : 2 ports
 - ftp 21 = contrôle
 - ftp-data 20 = transfert de données

Modèle d'interaction FTP



root@broceliande(3): /usr/etc/etherfind -v -proto tcp between firmin galere

chevrier@galere(1): ftp firmin

"etablissement de la connexion tcp"

TCP from galere.2843 to firmin.ftp seq 4BF83E00, SYN

TCP from firmin.ftp to galere.2843 seq 3940400, ack 4BF83E01, SYN

TCP from galere.2843 to firmin.ftp seq 4BF83E01, ack 3940401

Connected to firmin.

TCP from firmin.ftp to galere.2843 seq 3940401, ack 4BF83E01, 42 bytes data

220 firmin FTP server (SunOS 4.1) ready.

TCP from galere.2843 to firmin.ftp seq 4BF83E01, ack 394042B

Name (firmin:chevrier):

TCP from galere.2843 to firmin.ftp seq 4BF83E01, ack 394042B, 15 bytes data

USER chevrier

TCP from firmin.ftp to galere.2843 seq 394042B, ack 4BF83E10,

TCP from firmin.ftp to galere.2843 seq 394042B, ack 4BF83E10, 37 bytes data

331 Password required for chevrier.

TCP from galere.2843 to firmin.ftp seq 4BF83E10, ack 3940450,

Password:

TCP from galere.2843 to firmin.ftp seq 4BF83E10, ack 3940450, 15 bytes data

PASS mot_de_passe

TCP from firmin.ftp to galere.2843 seq 3940450, ack 4BF83E1F

TCP from firmin.ftp to galere.2843 seq 3940450, ack 4BF83E1F, 30 bytes data

230 User chevrier logged in.

TCP from galere.2843 to firmin.ftp seq 4BF83E1F, ack 394046E

ftp> dir

TCP from galere.2843 to firmin.ftp seq 4BF83E1F, ack 394046E, 26 bytes data

PORT galere,2844

TCP from firmin.ftp to galere.2843 seq 394046E, ack 4BF83E39

TCP from firmin.ftp to galere.2843 seq 394046E, ack 4BF83E39, 30 bytes data

200 PORT command successful.

TCP from galere.2843 to firmin.ftp seq 4BF83E39, ack 394048C, 6 bytes data

LIST

TCP from firmin.ftp to galere.2843 seq 394048C, ack 4BF83E3F

"établissement d'une connexion donnée"

TCP from firmin.ftp-data to galere.2844 seq 3AE6200, SYN,

TCP from galere.2844 to firmin.ftp-data seq 4C139600, ack 3AE6201, SYN,

TCP from firmin.ftp-data to galere.2844 seq 3AE6201, ack 4C139601,

TCP from firmin.ftp to galere.2843 seq 394048C, ack 4BF83E3F, 71 bytes data

150 ASCII data connection for /bin/ls (147.210.8.171,2844) (0 bytes).

TCP from galere.2843 to firmin.ftp seq 4BF83E3F, ack 39404D3,

TCP from firmin.ftp to galere.2843 seq 39404D3, ack 4BF83E3F, 30 bytes data

226 ASCII Tranfer completed

TCP from galere.2843 to firmin.ftp seq 4BF83E3F, ack 39404F1,

TCP from firmin.ftp-data to galere.2844 seq 3AE6201,

ack 4C139601,1460 bytes data

TCP from firmin.ftp-data to galere.2844 seq 3AE831D,

ack 4C139601, FIN, 1354 bytes data

TCP from galere.2844 to firmin.ftp-data seq 4C139601, ack 3AE8868,

TCP from galere.2844 to firmin.ftp-data seq 4C139601, ack 3AE8868, FIN

TCP from firmin.ftp-data to galere.2844 seq 3AE8868, ack 4C139602,

ftp>bye

TCP from galere.2843 to firmin.ftp seq 4BF83E8F, ack 3940617, 6 bytes data

QUIT

TCP from firmin.ftp to galere.2843 seq 3940617, ack 4BF83E95, 14 bytes data

221 Goodbye.

« Fermeture de la connexion »

TCP from galere.2843 to firmin.ftp seq 4BF83E95, ack 3940625, FIN,

TCP from firmin.ftp to galere.2843 seq 3940625, ack 4BF83E96

TCP from firmin.ftp to galere.2843 seq 3940625, ack 4BF83E96, FIN,

TCP from galere.2843 to firmin.ftp seq 4BF83E96, ack 3940626

Le prochain TD (machine)

1. Client FTP : votre navigateur
2. Client FTP en 'ligne de commandes'
3. Spécification de FTP : RFC 959
4. Dialogue avec un serveur FTP en utilisant telnet
5. Client FTP en C