

Polytech' Grenoble – RICM 1^{ère} année

mercredi 12 mai 2004

EXAMEN DE RESEAUX

Durée 3 heures
Tous documents autorisés
Les 3 questions sont indépendantes.

EXERCICE 1 (9 POINTS) ETUDE DU PROTOCOLE TFTP

Le protocole TFTP (Trivial File Transfert Protocol) est un protocole de transfert de fichiers simple utilisé en particulier dans les phases de configuration de machines depuis un serveur.

Il est développé au-dessus du protocole de transfert UDP.

Fonctionnement :

Le client envoie dans un premier message le nom du fichier et le sens du transfert. Le serveur n'acquiesce pas ce paquet. Le serveur envoie/reçoit ensuite le fichier par blocs de 512 octets. Un numéro de blocs (de 1 à n) est envoyé avec chacun de ces paquets. Un bloc de moins de 512 octets indique la fin du fichier.

Le récepteur doit envoyer un acquiescement pour chaque paquet. L'émetteur n'envoie le bloc suivant qu'à la réception de l'acquiescement du bloc précédent. L'acquiescement comporte le numéro du bloc qu'il acquiesce.

Un temporisateur est associé à chaque paquet de données et aux acquiescements. En cas de sonnerie de ce timer, le paquet ou l'acquiescement correspondant est réémis. Le traitement en cas de réception d'un paquet de données ou un acquiescement déjà reçu (doublons) provoque le même traitement (émission d'un acquiescement dans le premier cas ou émission du paquet suivant dans le second).

Un message d'erreur peut être émis en cas de problème d'incohérence dans le transfert (perte du premier paquet, rémissions successives trop nombreuses ...). Le transfert est abandonné en cas de réception d'un message d'erreur.

Format des paquets TFTP de données et d'acquittement :

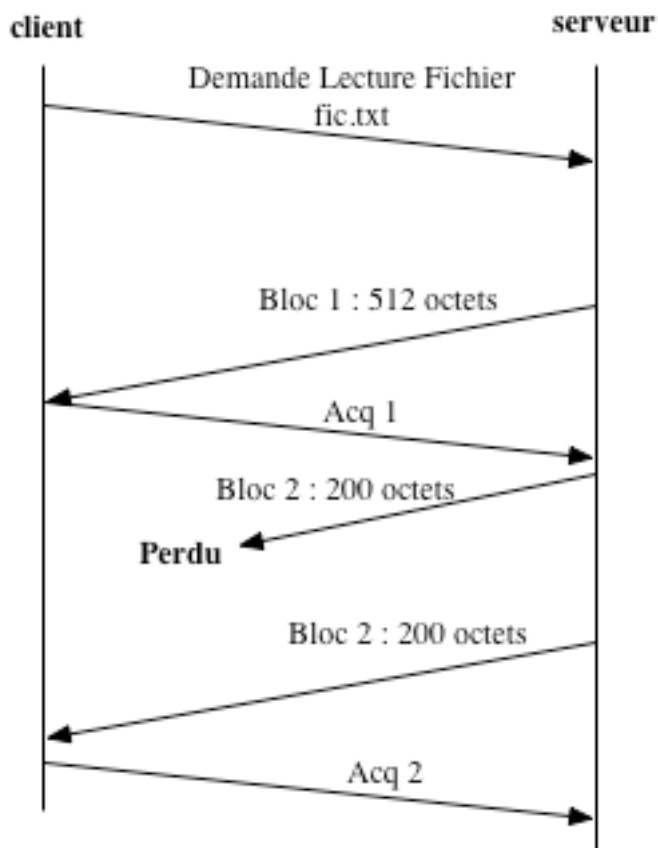
Bloc de donnée

Type = 03 (2 octets)	Numéro de bloc (2 octets)	Données (0 à 512 octets)
-------------------------	------------------------------	--------------------------

Acquittement

Type = 04 (2 octets)	Numéro de bloc (2 octets)
-------------------------	------------------------------

1- Soit l'échange suivant :

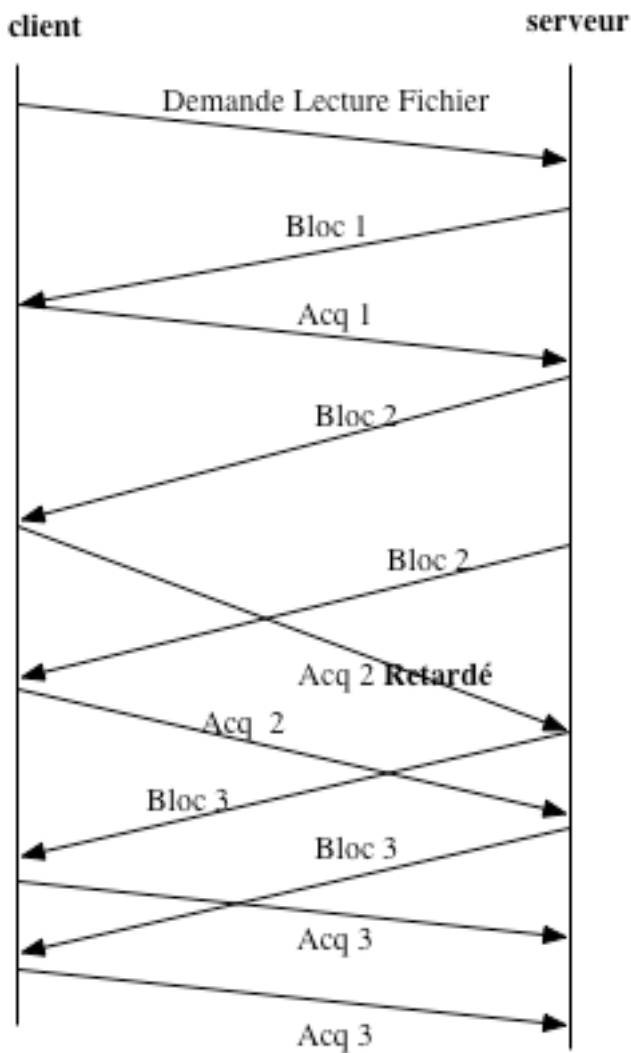


a- Quelle est la taille du fichier transféré ?

b- Pourquoi le Bloc 2 est-il réémis ?

c- Dans quel cas peut-on avoir l'émission d'un bloc de taille nulle ?

2- Soit l'échange donné ci-dessous.
L'acquittement du bloc 2 a été retardé.



- a- Pourquoi le bloc 2 est-il réémis par le server ?
- b- Pourquoi l'acquittement du bloc 3 est-il réémis ?
- c- Continuez l'échange dans le cas où il y aurait un bloc 4 et 5 à transférer.

3- Voici l'algorithme du récepteur une fois le premier paquet transféré (comportant le nom du fichier et le sens de transfert).

Numero=1 ;

Répéter

Suivant événement faire

Si Reception (paquet) alors

Numero= Numero-bloc (paquet)

Composer-fichier (Contenu-Fichier, Numero, Donnee-bloc (paquet),

Taille-bloc (paquet))

Emettre (acquittance, Numero)

Armer-timer

Si sonnerie timer

Emettre acquittance (Numero)

Armer-timer

Tant que Taille-bloc (paquet) = 512

La procédure **Numero-bloc** retourne le numéro du bloc figurant dans le paquet TFTP.

La fonction **Taille-bloc** retourne la taille du bloc contenu dans un paquet.

Armer-timer lance un timer de durée fixe. Dans le cas où ce timer est déjà en cours, il est relancé avec sa valeur d'origine.

La procédure **Reception** permet de récupérer un paquet reçu depuis le réseau.

La procédure **Emettre** permet d'émettre sur le réseau (passage au protocole de la couche inférieure) un paquet qui portera le numéro donné.

La fonction **Donnee-bloc** renvoie les données du bloc contenu dans un paquet TFTP.

La procédure **Composer-fichier** construit le contenu du fichier à partir du numéro du bloc, de sa taille et de son contenu.

On ne tient pas compte ici de la phase d'abandon en cas d'erreur.

Le fichier est ensuite écrit sur le disque s'il comporte tous les blocs. Dans le cas contraire un message d'erreur est envoyé à l'émetteur.

a- Donnez l'algorithme de l'émetteur sous une forme équivalente.

On supposera avoir à disposition la procédure **Extraire-Bloc(Fichier, Numéro, Bloc, Taille)** qui renvoie à partir du fichier **Fichier**, le contenu du bloc **Bloc** de numéro **Numéro** ainsi que sa taille (**Bloc** et **Taille** sont des paramètres résultats). On utilisera aussi les procédures **Emettre**, **Reception** et **Numero-Bloc**.

b- Que se passe-t-il si le dernier acquittance se perd ?

c- Que se passe t-il si deux paquets de données arrivent dans le désordre ? On donnera un scénario pour illustrer ses explications.

4- On utilise ce protocole sur un réseau Ethernet à 10 mégabit/s.

On veut estimer le débit observé au niveau de l'application dans le cas de l'utilisation du réseau lors d'un transfert d'un gros fichier. On suppose qu'il n'y a pas de perte/erreur de paquet et qu'aucun autre échange de paquet n'est effectué sur le réseau

Pour cette estimation on tiendra compte :

- Du protocole TFTP
- De la taille des entêtes des protocoles sous jacents (UDP : 8 octets, IP : 20 octets, Ethernet 18 octets, 8 octets de préambule physique, 9,6 micro secondes de silence Inter trame) ainsi que du format des paquets TFTP.
- Du délai de propagation du réseau qui est supposé de 20 microsecondes (10 microsecondes entre les machines et le switch ou le hub).

a- Donnez une estimation du débit dans les deux cas suivants :

- Utilisation d'un concentrateur(ou Hub)
- Utilisation d'un commutateur (switch) qui ne réémet les trames que lorsqu'il les a complètement reçues. On supposera nul le temps d'attente/traitement dans le commutateur.

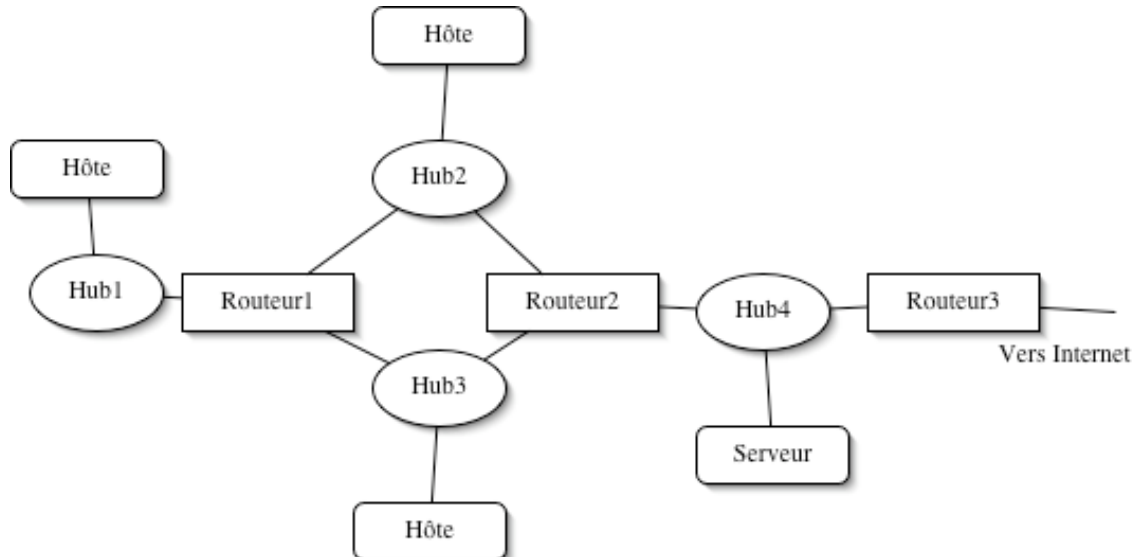
b- Estimez à nouveau ces débits dans le cas de l'utilisation d'une autre application de transfert de fichier (comme FTP) qui utilise TCP et ne nécessite donc pas de mettre en place un mécanisme de récupération d'erreur au niveau « applicatif ». On supposera que lors du transfert du contenu du fichier, l'application passe à TCP le contenu du fichier en une seule fois. On négligera les acquittements TCP.

On expliquera ici l'intérêt du mécanisme à fenêtre d'anticipation de TCP.

On donnera pour expliquer ces différents calculs des figures faisant apparaître les délais d'émissions et de propagations.

EXERCICE 2 (8 POINTS) ADMINISTRATION

Vous avez à administrer un réseau comportant la topologie suivante:



Les 4 Hubs sont de type Ethernet.

Les routeurs possèdent des interfaces réseau Ethernet. Le routeur 3 possède une interface réseau supplémentaire d'un autre type permettant de se raccorder au monde extérieur (Internet).

Les Hôtes sont des machines utilisateurs, ils sont au nombre de 10 à être connectés sur chacun des hubs 1, 2 et 3. Le serveur est un ordinateur plus puissant utilisé comme serveur de fichiers et serveur WEB.

Vous disposez d'une plage d'adresses Internet pour configurer vos réseaux locaux: **60.0.0.0/26**, («/x» indique le nombre de bits de la partie réseau de l'adresse).

L'adresse du routeur 3 vers l'extérieur doit être **192.0.0.1**.

Le premier routeur d'Internet qui est connecté au routeur 3 possède l'adresse **192.0.0.2**.

1- Proposez un plan d'adressage pour cet Intranet. On pourra rendre l'ANNEXE annotée des adresses.

Donnez le Netmask associé à chacune des 4 adresses Réseau utilisées.

Combien de machines utilisateurs pouvez-vous connecter au maximum sur chacun des 3 réseaux Ethernet ?

Donner les plages d'adresses machines pour chaque réseau.

2- Dans le cas où l'on veut connecter 25 machines au hub1, proposez à nouveau des adresses pour chacun des réseaux de l'Intranet. Combien de machines peut-on alors connecter à chacun des réseaux ?

On ne redonnera pas le plan d'adressage complet de l'Intranet.

3- Dans la suite on suppose avoir mis en place le plan d'adressage donné à la question 1. On pourra définir des alias pour les adresses réseaux et machines et les utiliser dans les tables de routage.

Donnez le contenu des tables de routage des routeurs, des machines utilisateur et du serveur pour que toutes les machines utilisateur et le serveur puissent accéder à l'ensemble de l'Intranet et de l'Internet .

On donnera les tables de routage sous la forme :

Réseau destination / Adresse du routeur voisin / Netmask

4- Donnez le contenu des tables de routage des routeurs (ou les modifications par rapport à la question précédente), des machines utilisateurs et du serveur afin de répondre aux contraintes suivantes :

- Les machines utilisateur connectées au Hub2 accèdent à l'Internet
- Les machines utilisateur connectées au Hub1 et 3 n'accèdent pas à l'Internet
- Toutes les machines utilisateurs de l'Intranet peuvent accéder aux autres machines utilisateur et au serveur.
- Le serveur peut accéder à l'Internet.
- Le passage du réseau du Hub2 au réseau du Hub3 et réciproquement se fait par le routeur 1

5- Le routeur 1 tombe en panne.

Comment remédier à ce problème pour que toutes les machines de votre intranet communiquent encore entre elles et avec l'extérieur ? On donnera la solution nécessitant le moins possible de changements de topologie et de modifications de configuration. Bien sûr vous n'avez pas à votre disposition de routeur de secours.

6- On lance le protocole RIP sur les 3 routeurs, les machines utilisateurs et le serveur.

Donnez le contenu des tables de routage des 3 routeurs et du serveur une fois qu'elles se sont stabilisées.

Donnez le contenu des paquets RIP envoyés par les routeurs 1 et 2 sur les réseaux des Hubs 2 et 3.

- 7- Dans les conditions de la question 7, le lien entre le routeur 2 et le hub 2 est cassé. Expliquez ce qu'il se passe et donnez le contenu des tables de routage résultantes pour les routeurs 1 et 2 ainsi que les machines utilisateurs connectées au hub 2 et 3.

EXERCICE 3 (3 POINTS)

L'annexe 2 contient les traces données par un outil de capture de messages (snoop sur Solaris) sur un réseau. La première trace donne une vision de 76 paquets de façon sommaire. La deuxième trace une vision des mêmes paquets (de 1 à 10, de 53 à 56 et de 73 à 76) dans un mode un peu plus détaillé (mode intermédiaire).

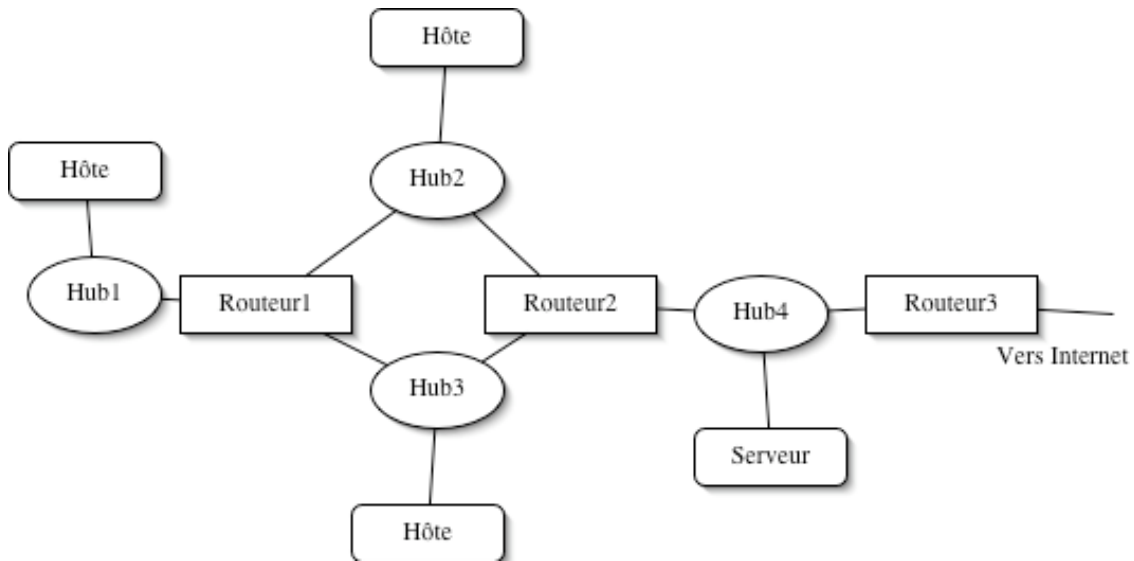
Remarques : En mode sommaire, la fin des lignes donne les caractères envoyés par l'application utilisant le « réseau » (Ex: le paquet 17 contient (au niveau des données de l'application) les codes ASCII des caractères: `{1}sparc2`). Il est possible que tous les caractères n'apparaissent pas par effet de troncature de la ligne. Le « retour à la ligne » n'est pas toujours visible. L'application **rlogin** est une application permettant de se connecter à distance sur une machine (similaire à Telnet).

Il s'agit de comprendre à quoi correspond la capture de l'ensemble de ces paquets. Pour vous aider, répondez attentivement à ces questions :

- 1- Quels sont les protocoles de niveaux liaison de données, réseau et transport pour lesquels les entêtes apparaissent dans les paquets 1 et 2 ? A quoi servent ces paquets ? Que veut dire « broadcast » ?
- 2- Sur quelle machine a-t-on lancé le RLOGIN (client) ? Les deux machines appartiennent-elles au même réseau local ? Pourquoi ?
- 3- A quoi correspondent les paquets 3, 4 et 5 ?
Expliquez à quoi servent les champs **destination port** et **source port** (**D=** et **S=** en mode intermédiaire) dans l'entête TCP.
- 4- Pourquoi le champs **Ack** de l'entête TCP du paquet 9 est-il égal à 1427616695 ?
Ce paquet contient-il des données de l'application ?
- 5- Quelles données contiennent les paquets 53, 54, 55 et 56 ?

6- Quelles commandes Unix ont été tapées par l'utilisateur qui a lancé le RLOGIN ?
Pourquoi les caractères tapés pendant le rlogin circulent deux fois sur le réseau (sauf pour le mot de passe) ?

ANNEXE 1 :



ANNEXE 2 :

Traces en mode sommaire :

```
1 0.000 192.0.0.129 -> (broadcast) ARP C Who is 192.0.0.130, 192.0.0.130 ?
2 0.00021 192.0.0.130 -> 192.0.0.129 ARP R 192.0.0.130, 192.0.0.130 is 0:b0:d0:5c:5b:37
3 0.00007 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
4 0.00020 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023
5 0.00008 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
6 0.00010 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
7 0.00011 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023
8 0.00008 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023 root\0root\0sun-cmd/96
9 0.00015 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023
10 0.00999 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023
11 0.00005 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
12 0.01753 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023
13 0.00010 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
14 0.00038 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
15 0.00467 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023 Password:
16 0.04117 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
17 1.29710 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023 r
18 0.04087 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023
19 0.17864 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023 o
20 0.04133 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023
21 0.08922 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023 o
22 0.04077 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023
23 0.04173 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023 t
24 0.04826 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023
25 0.06067 192.0.0.130 -> 192.0.0.191 RIP R (1 destinations)
26 0.83576 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023 1
27 0.04356 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023
28 0.11462 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
29 0.00034 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023
30 0.04705 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
31 0.00020 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023 Last login: Fri Dec
32 0.04979 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
33 3.95289 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023 p
34 0.00023 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023 p
35 0.04687 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
```

```

36 0.38426 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023 w
37 0.00016 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023 w
38 0.04555 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
39 0.17671 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023 d
40 0.00016 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023 d
41 0.04312 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
42 1.13330 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
43 0.00018 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023
44 0.04650 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
45 0.00012 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023 ^r\n#
46 0.04987 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
47 2.18003 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023 l
48 0.00019 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023 l
49 0.04977 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
50 0.04887 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023 s
51 0.00015 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023 s
52 0.04096 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
53 0.45424 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
54 0.00019 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023
55 0.04556 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
56 0.00027 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023 ALIRE.beholder.insta
57 0.04973 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
58 0.23025 192.0.0.129 -> 192.0.0.191 RIP R (1 destinations)
59 2.02735 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023 e
60 0.00018 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023 e
61 0.04221 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
62 0.18465 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023 x
63 0.00015 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023 x
64 0.04518 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
65 0.15052 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023 i
66 0.00015 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023 i
67 0.04930 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
68 0.07294 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023 t
69 0.00015 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023 t
70 0.04689 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
71 0.35051 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
72 0.00017 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023
73 0.00414 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023
74 0.00006 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
75 0.00205 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
76 0.00012 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023

```

Traces en mode intermédiaire :

```

-----
1 0.00000 192.0.0.129 -> (broadcast) ETHER Type=0806 (ARP), size = 42 bytes
1 0.00000 192.0.0.129 -> (broadcast) ARP C Who is 192.0.0.130, 192.0.0.130 ?
-----
2 0.00021 192.0.0.130 -> 192.0.0.129 ETHER Type=0806 (ARP), size = 60 bytes
2 0.00021 192.0.0.130 -> 192.0.0.129 ARP R 192.0.0.130, 192.0.0.130 is
0:b0:d0:5c:5b:37
-----
3 0.00007 192.0.0.129 -> 192.0.0.130 ETHER Type=0800 (IP), size = 58 bytes
3 0.00007 192.0.0.129 -> 192.0.0.130 IP D=192.0.0.130 S=192.0.0.129 LEN=44,
ID=11317
3 0.00007 192.0.0.129 -> 192.0.0.130 TCP D=513 S=1023 Syn Seq=1427616670
Len=0 Win=8760 Options=<mss 1460>
3 0.00007 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
-----
4 0.00020 192.0.0.130 -> 192.0.0.129 ETHER Type=0800 (IP), size = 60 bytes
4 0.00020 192.0.0.130 -> 192.0.0.129 IP D=192.0.0.129 S=192.0.0.130 LEN=44,
ID=48593

```

```

4 0.00020 192.0.0.130 -> 192.0.0.129 TCP D=1023 S=513 Syn Ack=1427616671
Seq=1426852508 Len=0 Win=8760 Options=<mss 1460>
4 0.00020 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023

-----
5 0.00008 192.0.0.129 -> 192.0.0.130 ETHER Type=0800 (IP), size = 54 bytes
5 0.00008 192.0.0.129 -> 192.0.0.130 IP D=192.0.0.130 S=192.0.0.129 LEN=40,
ID=11318
5 0.00008 192.0.0.129 -> 192.0.0.130 TCP D=513 S=1023 Ack=1426852509
Seq=1427616671 Len=0 Win=8760
5 0.00008 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023

-----
6 0.00010 192.0.0.129 -> 192.0.0.130 ETHER Type=0800 (IP), size = 55 bytes
6 0.00010 192.0.0.129 -> 192.0.0.130 IP D=192.0.0.130 S=192.0.0.129 LEN=41,
ID=11319
6 0.00010 192.0.0.129 -> 192.0.0.130 TCP D=513 S=1023 Ack=1426852509
Seq=1427616671 Len=1 Win=8760
6 0.00010 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023

-----
7 0.00011 192.0.0.130 -> 192.0.0.129 ETHER Type=0800 (IP), size = 60 bytes
7 0.00011 192.0.0.130 -> 192.0.0.129 IP D=192.0.0.129 S=192.0.0.130 LEN=40,
ID=48594
7 0.00011 192.0.0.130 -> 192.0.0.129 TCP D=1023 S=513 Ack=1427616672
Seq=1426852509 Len=0 Win=8759
7 0.00011 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023

-----
8 0.00008 192.0.0.129 -> 192.0.0.130 ETHER Type=0800 (IP), size = 77 bytes
8 0.00008 192.0.0.129 -> 192.0.0.130 IP D=192.0.0.130 S=192.0.0.129 LEN=63,
ID=11320
8 0.00008 192.0.0.129 -> 192.0.0.130 TCP D=513 S=1023 Ack=1426852509
Seq=1427616672 Len=23 Win=8760
8 0.00008 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023 root\0root\0sun-cmd/96

-----
9 0.00015 192.0.0.130 -> 192.0.0.129 ETHER Type=0800 (IP), size = 60 bytes
9 0.00015 192.0.0.130 -> 192.0.0.129 IP D=192.0.0.129 S=192.0.0.130 LEN=40,
ID=48595
9 0.00015 192.0.0.130 -> 192.0.0.129 TCP D=1023 S=513 Ack=1427616695
Seq=1426852509 Len=0 Win=8760
9 0.00015 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023

-----
10 0.00999 192.0.0.130 -> 192.0.0.129 ETHER Type=0800 (IP), size = 60 bytes
10 0.00999 192.0.0.130 -> 192.0.0.129 IP D=192.0.0.129 S=192.0.0.130 LEN=41,
ID=48596
10 0.00999 192.0.0.130 -> 192.0.0.129 TCP D=1023 S=513 Ack=1427616695
Seq=1426852509 Len=1 Win=8760
10 0.00999 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023

-----
53 0.45424 192.0.0.129 -> 192.0.0.130 ETHER Type=0800 (IP), size = 55 bytes
53 0.45424 192.0.0.129 -> 192.0.0.130 IP D=192.0.0.130 S=192.0.0.129 LEN=41,
ID=11346
53 0.45424 192.0.0.129 -> 192.0.0.130 TCP D=513 S=1023 Ack=1426852641
Seq=1427616719 Len=1 Win=8760
53 0.45424 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023

-----
54 0.00019 192.0.0.130 -> 192.0.0.129 ETHER Type=0800 (IP), size = 60 bytes
54 0.00019 192.0.0.130 -> 192.0.0.129 IP D=192.0.0.129 S=192.0.0.130 LEN=42,
ID=48613
54 0.00019 192.0.0.130 -> 192.0.0.129 TCP D=1023 S=513 Ack=1427616720
Seq=1426852641 Len=2 Win=8760
54 0.00019 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023

-----
55 0.04556 192.0.0.129 -> 192.0.0.130 ETHER Type=0800 (IP), size = 54 bytes

```

```

55 0.04556 192.0.0.129 -> 192.0.0.130 IP D=192.0.0.130 S=192.0.0.129 LEN=40,
ID=11347
55 0.04556 192.0.0.129 -> 192.0.0.130 TCP D=513 S=1023 Ack=1426852643
Seq=1427616720 Len=0 Win=8760
55 0.04556 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
-----
56 0.00027 192.0.0.130 -> 192.0.0.129 ETHER Type=0800 (IP), size = 1021 bytes
56 0.00027 192.0.0.130 -> 192.0.0.129 IP D=192.0.0.129 S=192.0.0.130 LEN=1007,
ID=48614
56 0.00027 192.0.0.130 -> 192.0.0.129 TCP D=1023 S=513 Ack=1427616720
Seq=1426852643 Len=967 Win=8760
56 0.00027 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023 ALIRE.beholder.insta
-----
73 0.00414 192.0.0.130 -> 192.0.0.129 ETHER Type=0800 (IP), size = 60 bytes
73 0.00414 192.0.0.130 -> 192.0.0.129 IP D=192.0.0.129 S=192.0.0.130 LEN=40,
ID=48620
73 0.00414 192.0.0.130 -> 192.0.0.129 TCP D=1023 S=513 Fin Ack=1427616725
Seq=1426853616 Len=0 Win=8760
73 0.00414 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023
-----
74 0.00006 192.0.0.129 -> 192.0.0.130 ETHER Type=0800 (IP), size = 54 bytes
74 0.00006 192.0.0.129 -> 192.0.0.130 IP D=192.0.0.130 S=192.0.0.129 LEN=40,
ID=11358
74 0.00006 192.0.0.129 -> 192.0.0.130 TCP D=513 S=1023 Ack=1426853617
Seq=1427616725 Len=0 Win=8760
74 0.00006 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
-----
75 0.00205 192.0.0.129 -> 192.0.0.130 ETHER Type=0800 (IP), size = 54 bytes
75 0.00205 192.0.0.129 -> 192.0.0.130 IP D=192.0.0.130 S=192.0.0.129 LEN=40,
ID=11359
75 0.00205 192.0.0.129 -> 192.0.0.130 TCP D=513 S=1023 Fin Ack=1426853617
Seq=1427616725 Len=0 Win=8760
75 0.00205 192.0.0.129 -> 192.0.0.130 RLOGIN C port=1023
-----
76 0.00012 192.0.0.130 -> 192.0.0.129 ETHER Type=0800 (IP), size = 60 bytes
76 0.00012 192.0.0.130 -> 192.0.0.129 IP D=192.0.0.129 S=192.0.0.130 LEN=40,
ID=48621
76 0.00012 192.0.0.130 -> 192.0.0.129 TCP D=1023 S=513 Ack=1427616726
Seq=1426853617 Len=0 Win=8760
76 0.00012 192.0.0.130 -> 192.0.0.129 RLOGIN R port=1023

```