

mode

C'est le mode du transfert de fichier : "netascii", "octet", ou "mail", comme défini dans la [RFC1350]. C'est un champ qui se termine par un NULL.

opt1

C'est la première option, en ASCII insensible à la casse (par exemple, "blksize"). C'est un champ ASCII terminé par un NUL.

valeur1

La valeur associée à la première option, en ASCII insensible à la casse. C'est un champ qui se termine par un NUL.

optN, valeurN

La paire option/valeur finale. Chaque champ terminé par un NUL est spécifié en ASCII insensible à la casse.

Les options et valeurs sont toutes terminées par NUL, en conservant le format de la demande d'origine. Si plusieurs options doivent être négociées, elles sont ajoutées les unes à la suite des autres. L'ordre dans lequel les options sont spécifiées n'est pas significatif. La taille maximum d'un paquet de demande est 512 octets.

Le paquet OACK a le format suivant :

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|  opc  |  opt1  |  0  | valeur1 |  0  |  optN  |  0  | valeurN |  0  |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

opc

Le champ opcode contient un 6, pour Accusé de réception d'option .

opt1

C'est le premier accusé de réception d'option, copié de la demande d'origine.

valeur1

C'est la valeur de l'accusé de réception associée à la première option. Si, et comment cette valeur peut différer de la demande d'origine est précisé dans la spécification de l'option.

optN, valeurN

C'est l'accusé de réception de la paire d'option/valeur finale.

Protocole de négociation

Le client ajoute les options à la fin du paquet Demande de lecture ou Demande d'écriture, comme montré ci-dessus. N'importe quel nombre d'options peut être spécifié ; cependant, une option ne peut être spécifiée qu'une seule fois. L'ordre des options n'est pas significatif.

Si le serveur prend en charge la négociation d'options, et si il reconnaît une ou plusieurs des options spécifiées dans le paquet de demande, le serveur peut répondre par un Accusé de réception d'options (OACK). Chaque option que reconnaît le serveur, et dont il accepte la valeur, est incluse dans le OACK. Certaines options peuvent permettre que plusieurs valeurs soient proposées, mais cette disposition est spécifique d'une option. Le serveur ne doit pas inclure dans le OACK d'option qui n'ait pas été spécifiquement demandée par le client ; c'est-à-dire que seul le client peut initier la négociation d'options. Les options que le serveur n'accepte pas devraient être omises de l'OACK ; elle ne doivent pas causer la génération d'un paquet ERREUR. Si la valeur d'une option prise en charge est invalide, la spécification pour cette option va indiquer si le serveur devrait simplement omettre l'option de l'OACK, répondre par une valeur de remplacement, ou envoyer un paquet ERREUR, avec le code d'erreur 8, pour terminer le transfert.

Une option non acquittée par le serveur doit être ignorée par le client et le serveur comme si elle n'avait jamais été demandée. Si plusieurs options étaient demandées, le client doit utiliser les options qui ont été acquittées par le serveur et ne doit pas utiliser les options qui n'ont pas été acquittées par le serveur.

Lorsque le client ajoute des options à la fin d'un paquet Demande de lecture, trois réponses possibles peuvent être retournées par le serveur :

OACK – accuse réception de la Demande de lecture et des options ;

DATA – accuse réception de la demande de lecture, mais pas des options ;

ERREUR – la demande est refusée.

Lorsque le client ajoute des options à la fin d'un paquet Demande d'écriture, trois réponses possibles peuvent être retournées par le serveur :

OACK – accuse réception de la demande d'écriture et des options ;
 ACK - accuse réception de la demande d'écriture, mais pas des options ;
 ERREUR – la demande est refusée.

Si une mise en œuvre de serveur ne prend pas en charge la négociation d'option, elle va vraisemblablement ignorer toutes les options ajoutées à la demande du client. Dans ce cas, le serveur va retourner un paquet DATA pour une demande de lecture et un paquet ACK pour une demande d'écriture, établissant un transfert de données TFTP normal. Dans le cas où un serveur retourne une erreur pour une demande qui porte une option, le client peut tenter de répéter la demande sans ajouter aucune option. Cette option de mise en œuvre va servir aux serveurs qui considèrent les données étrangères dans le paquet de demande comme erronées.

Selon la demande originale de transfert, il y a deux façons pour qu'un client confirme son acceptation du OACK d'un serveur. Si le transfert a été initié avec une demande d'écriture, un ACK (avec le numéro de bloc de données réglé à 0) est alors envoyé par le client pour confirmer les valeurs dans le paquet OACK du serveur. Si le transfert a été initié avec une demande d'écriture, le client commence alors le transfert avec le premier paquet DATA, en utilisant les valeurs négociées. Si le client rejette le OACK, il envoie alors au serveur un paquet ERREUR, avec le code d'erreur 8, et le transfert est terminé.

Une fois qu'un client a accusé réception d'un OACK, avec une réponse appropriée de non erreur, ce client a accepté d'utiliser seulement les options et valeurs retournées par le serveur. On se souviendra que le serveur ne peut pas demander une option ; il peut seulement y répondre. Si le client reçoit un OACK qui contient une option non reconnue, il devrait répondre par un paquet ERREUR, avec le code d'erreur 8, et terminer le transfert.

Exemples

Demande d'écriture

client	serveur
1 foofile 0 octet 0 blksize 0 1432 0 -->	RRQ
4 0 -->	<-- 6 blksize 0 1432 0 OACK
4 1 -->	ACK
4 2 -->	<-- 3 1 1432 octets of data DATA
4 3 -->	ACK
4 4 -->	<-- 3 2 1432 octets of data DATA
4 5 -->	ACK
4 6 -->	<-- 3 3 <1432 octets of data DATA
4 7 -->	ACK

Demande d'écriture

client	serveur
2 barfile 0 octet 0 blksize 0 2048 0 -->	RRQ
3 1 2048 octets of data -->	<-- 6 blksize 0 2048 0 OACK
3 2 2048 octets of data -->	DATA
3 3 <2048 octets of data -->	<-- 4 1 ACK
3 4 <2048 octets of data -->	DATA
3 5 <2048 octets of data -->	<-- 4 2 ACK
3 6 <2048 octets of data -->	DATA
3 7 <2048 octets of data -->	<-- 4 3 ACK

Considérations pour la sécurité

Les questions de sécurité ne sont pas discutées dans le présent mémoire.

Références

- [RFC1350] K. Sollins, "[Protocole TFTP](#) (révision 2)", STD 33, juin 1992. (*MàJ par 1782-85, 2347_49*)
- [RFC1783] G. Malkin, A. Harkin, "Option TFTP Taille de bloc", mars 1995. (*P.S.*)
- [RFC1784] G. Malkin, A. Harkin, "Options TFTP Intervalle de fin de temporisation et Taille de transfert", mars 1995. (*P.S.*)

Adresse des auteurs

Gary Scott Malkin
Xylogics, Inc.
53 Third Avenue
Burlington, MA 01803
téléphone : (617) 272-8140
mél : gmalkin@xylogics.com

Art Harkin
Internet Services Project
Information Networks Division
19420 Homestead Road MS 43LN
Cupertino, CA 95014
téléphone : (408) 447-3755
mél : ash@cup.hp.com