

Groupe de travail Réseau
Request for Comments : 2347
 RFC mise à jour : 1350
 RFC rendue obsolète : 1782
 Catégorie : En cours de normalisation

G. Malkin, Bay Networks
 A. Harkin, Hewlett Packard Co.
 mai 1998

Traduction Claude Brière de L'Isle

Extension d'option TFTP

Statut de ce mémoire

Le présent document spécifie un protocole Internet en cours de normalisation pour la communauté de l'Internet, et appelle à des discussions et des suggestions pour son amélioration. Prière de se reporter à l'édition actuelle du STD 1 "Normes des protocoles officiels de l'Internet" pour connaître l'état de normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

Notice de copyright

Copyright (C) The Internet Society (1998).

Résumé

Le protocole trivial de transfert de fichier [RFC1350] est un simple protocole de transfert de fichier, en mode rigide (*lock-step*) qui permet à un client d'obtenir ou de mettre un fichier sur un hôte distant. Le présent document décrit une simple extension à TFTP pour permettre une négociation d'option avant le transfert de fichier.

Introduction

Le mécanisme de négociation d'option proposé dans le présent document est une extension rétro compatible au protocole TFTP. Il permet que des options de transfert de fichier soient négociées avant le transfert en utilisant un mécanisme qui est cohérent avec le format de paquet de demande de TFTP. Le mécanisme reste simple en appliquant une séquence de demande-réponse-accusé de réception, similaire à l'approche en mode rigide retenue par TFTP lui-même.

Alors que le mécanisme de négociation d'option est d'utilisation générale, en ce que de nombreux types d'options peuvent être négociés, il a été créé pour prendre en charge l'option Taille de bloc (*Blocksize*) définie dans la [RFC2348]. Des options supplémentaires sont définies dans la [RFC2349].

Formats de paquet

Les options TFTP sont ajoutées aux paquets Demande de lecture et Demande d'écriture. Un nouveau type de paquet TFTP, l'accusé de réception d'option (OACK, *Option Acknowledgment*) est utilisé pour accuser réception de la demande de négociation d'option d'un client. Un nouveau code d'erreur, 8, est aussi défini pour indiquer qu'un transfert devrait être terminé du fait de la négociation d'option.

Les options sont ajoutées comme suit à un paquet Demande de lecture ou Demande d'écriture TFTP :

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|opcode|filename| 0 | mode  | 0 | opt1  | 0 |valeur1| 0 | <
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
<
>-----+-----+-----+-----+
< optN  | 0 |valeurN| 0 |
>-----+-----+-----+-----+
```

opcode

Le champ opcode contient soit 1, pour les demandes de lecture, soit 2, pour les demandes d'écriture, comme défini dans la [RFC1350].

filename

C'est le nom du fichier à lire ou écrire, comme défini dans la [RFC1350]. C'est un champ terminé par NULL.

mode

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

opt1

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

valeur1

C'est la valeur associée à la première option, en ASCII insensible à la casse. C'est un champ terminé par un NULL.

optN, valeurN

C'est la paire option/valeur finale. Chaque champ terminé par un NULL est spécifié en ASCII insensible à la casse.

Les options et valeurs se terminent toutes par NULL, conformément au format de demande original. Si plusieurs options sont à négocier, elles sont ajoutées les unes aux autres. L'ordre dans lequel les options sont spécifiées n'est pas significatif. La taille maximum d'un paquet de demande est de 512 octets.

Le paquet OACK a le format suivant :

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

opcode

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

opt1

Accusé de réception de la première option, copiée de la demande originale.

valeur1

Valeur associée à la première option, dont il est accusé réception. Si et comment cette valeur peut différer de la demande originale est précisé dans la spécification de l'option.

optN, valeurN

Paire d'accusé de réception final d'option/valeur.

Protocole de négociation

Le client ajoute les options à la fin du paquet de demande de lecture ou d'écriture, comme montré ci-dessus. Tout 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 accepte la négociation d'option, 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, *Options Acknowledgment*). Chaque option que reconnaît le serveur, et dont il accepte la valeur, est incluse dans le OACK. Certaines options peuvent permettre que d'autres valeurs soient proposées, mais ceci est une caractéristique spécifique de l'option. Le serveur ne doit pas inclure dans le OACK d'option qui n'a pas été spécifiquement demandée par le client ; c'est-à-dire que seul le client peut initier la négociation d'option. Les options que le serveur n'accepte pas devraient être omises de l'OACK ; elles ne devraient 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 avec une autre valeur, 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 ont été demandées, le client doit utiliser celles 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 ;
 ERROR – la demande a été 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 ;
 ERROR – la demande a été refusée.

Si une mise en œuvre de serveur ne prend pas en charge la négociation d'option, elle va probablement 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 être utile aux serveurs qui considèrent que les données étrangères dans le paquet de demande sont erronées.

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

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

Exemples

Demande en lecture

```

client
-----
|1|foofile|0|octet|0|blksize|0|1432|0|  -->
                                <-- |6|blksize|0|1432|0|
|4|0|  -->                                ACK
                                <-- |3|1| 1432 octets de données | DATA
|4|1|  -->                                ACK
                                <-- |3|2| 1432 octets de données | DATA
|4|2|  -->                                ACK
                                <-- |3|3|<1432 octets de données | DATA
|4|3|  -->                                ACK

```

Demande en écriture

```

client
-----
|2|barfile|0|octet|0|blksize|0|2048|0|  -->
                                <-- |6|blksize|0|2048|0|
|3|1| 2048 octets de données |  -->                                DATA
                                <-- |4|1|                                ACK
|3|2| 2048 octets de données |  -->                                DATA
                                <-- |4|2|                                ACK
|3|3|<2048 octets de données |  -->                                DATA
                                <-- |4|3|                                ACK

```

Considérations sur la sécurité

Le protocole TFTP de base n'a pas de mécanisme de sécurité. C'est pourquoi il n'a pas de capacité de renommage, suppression ou écrasement de fichier. Le présent document n'apporte aucune sécurité à TFTP ; cependant, les extensions spécifiées n'ajoutent aucun risque supplémentaire pour la sécurité.

Références

- [RFC1350] K. Sollins, "[Protocole TFTP](#) (révision 2)", STD 33, juin 1992. (*MàJ par 1782-85, 2347_49*)
- [RFC2348] G. Malkin, A. Harkin, "[Option TFTP Taille de bloc](#)", mai 1998. (*D.S.*)
- [RFC2349] G. Malkin, A. Harkin, "Options TFTP [Intervalle de temporisation et taille de transfert](#)", mai 1998. (*D.S.*)

Adresse des auteurs

Gary Scott Malkin
Bay Networks
8 Federal Street
Billerica, MA 01821
téléphone : (978) 916-4237
mél : gmalkin@baynetworks.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

Déclaration complète de droits de reproduction

Copyright (C) The Internet Society (1998). Tous droits réservés.

Ce document et les traductions de celui-ci peuvent être copiés et diffusés, et les travaux dérivés qui commentent ou expliquent autrement ou aident à sa mise en œuvre peuvent être préparés, copiés, publiés et distribués, partiellement ou en totalité, sans restriction d'aucune sorte, à condition que l'avis de droits de reproduction ci-dessus et ce paragraphe soient inclus sur toutes ces copies et œuvres dérivées. Toutefois, ce document lui-même ne peut être modifié en aucune façon, par exemple en supprimant le droit d'auteur ou les références à l'Internet Society ou d'autres organisations Internet, sauf si c'est nécessaire à l'élaboration des normes Internet, auquel cas les procédures pour les droits de reproduction définies dans les processus des normes pour l'Internet doivent être suivies, ou si nécessaire pour le traduire dans des langues autres que l'anglais.

Les permissions limitées accordées ci-dessus sont perpétuelles et ne seront pas révoquées par la Société Internet, ses successeurs ou ayants droit.

Ce document et les renseignements qu'il contient sont fournis "TELS QUELS" et l'INTERNET SOCIETY et l'INTERNET ENGINEERING TASK FORCE déclinent toute garantie, expresse ou implicite, y compris mais sans s'y limiter, toute garantie que l'utilisation de l'information ici présente n'enfreindra aucun droit ou aucune garantie implicite de commercialisation ou d'adaptation à un objet particulier.

Remerciement

Le financement de la fonction d'éditeur des RFC est actuellement fourni par la Internet Society.