Groupe de travail Réseau
Request for Comments : 2428
Catégorie : En cours de normalisation
Traduction Claude Brière de L'Isle

M. Allman, NASA Lewis/Sterling Software
S. Ostermann, Ohio University
C. Metz, The Inner Net
septembre 1998

Extensions à FTP pour IPv6 et les NAT

Résumé
La spécification du protocole de transfert de fichier suppose que le protocole réseau sous-jacent utilise une adresse réseau de 32 bits (précisément, IP version 4). Avec le déploiement de la version 6 du protocole Internet, les adresses réseau ne vont plus être de 32 bits. Le présent article spécifie des extensions à FTP qui permettent au protocole de fonctionner sur IPv4 et IPv6. De plus, le cadre défini peut prendre en charge des protocoles réseau supplémentaires à l'avenir.

1. Introduction

Les mots clés tels que DOIT et DEVRAIT, qu'on trouve dans ce document sont utilisés comme défini dans la [RFC2119].

Le protocole de transfert de fichier [RFC0959] ne fournit que la capacité à communiquer des informations sur les connexions de données IPv4. FTP suppose que les adresses réseau feront 32 bits. Cependant, avec le déploiement de la version 6 du protocole Internet [RFC1883] les adresses ne feront plus 32 bits. La [RFC1639] spécifie des extensions à FTP pour permettre son utilisation sur divers protocoles réseau. Malheureusement, le mécanisme peut échouer dans un environnement multi protocoles. Durant la transition entre IPv4 et IPv6, FTP a besoin d'être capable de négocier le protocole réseau qui va être utilisé pour le transfert des données.

Le présent document donne une spécification d'un moyen pour que FTP puisse communiquer des informations de point d'extrémité de connexion de données pour des protocoles réseau autres que IPv4. Dans cette spécification, les commandes FTP PORT et PASV sont remplacées respectivement par EPRT et EPSV. Le présent document est organisé comme suit. La Section 2 présente la commande EPRT et la Section 3 présente la commande EPSV. La Section 4 définit l'utilisation de ces deux nouvelles commandes FTP. La Section 5 présente brièvement les considérations pour la sécurité. Finalement, la Section 6 donne des conclusions.

2. Commande EPRT

La commande EPRT permet de spécifier une adresse étendue pour la connexion de données. L'adresse étendue DOIT comporter le protocole réseau ainsi que les adresses réseau et de transport. Le format de EPRT est :

EPRT<espace><d><protocole-réseau><d><adresse-réseau><d><accès-tcp><d>

Le mot-clé de commande EPRT DOIT être suivi par une seule espace (ASCII 32). Suivant l'espace, un caractère délimiteur (<d>) DOIT être spécifié. Le caractère délimiteur DOIT être un des caractères ASCII dans la gamme de 33 à 126 inclus. Le caractère "|" (ASCII 124) est recommandé sauf si il coïncide avec un caractère nécessaire pour coder l'adresse réseau.

L'argument <protocole-réseau> DOIT être un numéro de famille d'adresse défini par l'IANA dans la dernière version des Numéros alloués (la [RFC1700] au moment de la rédaction du présent document). Ce numéro indique le protocole à utiliser (et, implicitement, la longueur de l'adresse). Le présent document utilise deux des numéros de famille d'adresse tirés de la [RFC1700] comme exemples, selon le tableau suivant :

Numéro de famille d'adresse	Protocole
1	Protocole Internet, version 4 [RFC0791]
2	Protocole Internet, version 6 [RFC1883]

L'argument <adresse-réseau> est une représentation de chaîne spécifique du protocole de l'adresse réseau. Pour les deux familles d'adresses spécifiées ci-dessus (numéros de famille d'adresse 1 et 2) les adresses DOIVENT avoir le format suivant :

Numéro AF	Format d'adresse	Exemple
1	décimal séparé par des points	132.235.1.2
2	représentations de chaîne IPv6 définies dans la [RFC2573]	1080::8:800:200C:417A

L'argument <accès-tcp> doit être la représentation en chaîne du numéro de l'accès TCP sur lequel l'hôte écoute la connexion de données.

Voici des exemple de commande EPRT :

EPRT |1|132.235.1.2|6275|

EPRT |2|1080::8:800:200C:417A|5282|

La première commande spécifie que le serveur devrait utiliser IPv4 pour ouvrir une connexion de données avec l'hôte "132.235.1.2" sur l'accès TCP 6275. La seconde commande spécifie que le serveur devrait utiliser le protocole réseau IPv6 et l'adresse réseau "1080::8:800:200C:417A" pour ouvrir la connexion de données TCP sur l'accès 5282.

À réception d'une commande EPRT valide, le serveur DOIT retourner un code de 200 (Commande OK). Les codes d'erreur négative standard 500 et 501 [RFC0959] sont suffisants pour traiter la plupart des erreurs (par exemple, les erreurs de syntaxe) impliquant la commande EPRT. Cependant, un code d'erreur supplémentaire est nécessaire. Le code de réponse 522 indique que le serveur ne prend pas en charge le protocole réseau demandé. L'interprétation de ce nouveau code d'erreur est :

5yz	Achèvement négatif
x2z	Connexions
xy2	Échec d'accès étendu – protocole réseau inconnu

La portion de texte de la réponse DOIT indiquer quels protocoles réseau le serveur prend en charge. Si le protocole réseau n'est pas accepté, le format de la chaîne de réponse DOIT être :

<texte déclarant que le protocole réseau n'est pas accepté> (prot1,prot2,...,protn)

Le code numérique spécifié ci-dessus et les informations de protocole entre les caractères '(' et ')' sont tous deux destinés à l'automate logiciel qui reçoit la réponse ; le message textuel entre le code numérique et le '(' est destiné à l'utilisateur humain et peut être tout texte arbitraire, mais NE DOIT PAS inclure les caractères '(' et ')'.Dans le cas ci-dessus, le texte DEVRAIT indiquer que le protocole réseau dans la commande EPRT n'est pas accepté par le serveur. La liste des protocoles à l'intérieur des parenthèses DOIT être une liste des numéros de famille d'adresses, séparés par des virgules. Voici deux exemples de chaînes de réponse :

Protocole réseau non accepté, utiliser (1)

Protocole réseau non accepté, utiliser (1,2)

3. Commande EPSV

La commande EPSV demande qu'un serveur écoute sur un accès de données et attende une connexion. La commande EPSV prend un argument facultatif. La réponse à cette commande ne comporte que le numéro d'accès TCP de la connexion qui écoute. Le format de la réponse est cependant similaire à l'argument de la commande EPRT. Cela permet d'utiliser les mêmes sous-programmes d'analyse pour les deux commandes. De plus, le format laisse un espace pour le protocole réseau et/ou l'adresse réseau, qui peut être nécessaire à l'avenir dans la réponse EPSV. Le code de réponse pour entrer un mode passif en utilisant une adresse étendue DOIT être 229. L'interprétation de ce code, selon la [RFC0959] est :

2yz	Achèvement positif
x2z	Connexions
xy9	Entré dans le mode passif étendu

Le texte retourné en réponse à la commande EPSV DOIT être :

<texte indiquant que le serveur est entré en mode passif étendu> (<d><d><d><accès-tcp><d>)

La portion de la chaîne comprise entre les parenthèses DOIT être la chaîne exacte nécessaire pour que la commande EPRT ouvre la connexion de données, comme spécifié ci-dessus.

Les deux premiers champs contenus entre les parenthèses DOIVENT être blancs. Le troisième champ DOIT être la représentation de chaîne du numéro d'accès TCP sur lequel écoute le serveur pour une connexion de données. Le protocole réseau utilisé par la connexion de données sera le même que celui utilisé par la connexion de contrôle. De plus, l'adresse réseau utilisée pour établir la connexion de données sera la même que celle utilisée pour la connexion de contrôle. Voici un exemple de chaîne de réponse :

Entrée en mode passif étendu (|||6446|)

Les codes standard d'erreur négative 500 et 501 sont suffisants pour traiter toutes les erreurs qui impliquent la commande EPSV (par exemple, des erreurs de syntaxe).

Lorsque la commande EPSV est produite sans argument, le serveur va choisir le protocole réseau pour la connexion de données sur la base du protocole utilisé pour la connexion de contrôle. Cependant, dans le cas de mandataire FTP, ce protocole pourrait n'être pas approprié pour la communication entre les deux serveurs. Donc, le client a besoin d'être capable de demander un protocole spécifique. Si le serveur retourne un protocole qui n'est pas accepté par l'hôte qui va se connecter à l'accès, le client DOIT produire une commande ABOR (interrompre) pour permettre au serveur de clore la connexion qui écoute. Le client peut alors envoyer une commande EPSV demandant l'utilisation d'un protocole réseau spécifique, comme suit :

EPSV<espace><protocole-réseau>

Si le protocole demandé est accepté par le serveur, il DEVRAIT utiliser ce protocole. Sinon, le serveur DOIT retourner le message d'erreur 522 comme mentionné à la Section 2.

Finalement, la commande EPSV peut être utilisée avec l'argument "ALL" pour informer les traducteurs d'adresse réseau que la commande EPRT (ainsi que d'autres commandes de données) ne sera plus utilisée. Voici un exemple de cette commande :

EPSV<espace>ALL

À réception d'une commande EPSV ALL, le serveur DOIT rejeter toutes les commandes d'établissement de connexion de données autres que EPSV (c'est-à-dire, EPRT, PORT, PASV, et autres). Cette utilisation de la commande EPSV est expliquée dans la Section 4.

4. Usage de la commande

Pour tous les transferts FTP où la ou les connexions de contrôle et de données sont établies entre les deux mêmes machines, la commande EPSV DOIT être utilisée. L'utilisation de la commande EPSV améliore les performances des transferts qui traversent des pare-feu ou des traducteurs d'adresse réseau (NAT, *Network Address Translator*). La [RFC1579] recommande d'utiliser la commande passive lorsque on est derrière des pare-feu car les pare-feu ne permettent généralement pas les connexions entrantes (qui sont requises avec l'utilisation de la commande PORT (EPRT)). De plus, l'utilisation de EPSV comme défini dans le présent document n'exige pas que les NAT changent l'adresse réseau dans le trafic lorsque il est transmis. Le NAT devrait changer l'adresse si la commande EPRT était utilisée. Finalement, si le client produit une commande "EPSV ALL", les NAT peuvent être capables de mettre la connexion sur un "chemin rapide" à travers le traducteur, car la commande EPRT ne sera jamais utilisée et donc, la traduction de la portion données des segments ne sera jamais nécessaire. Lorsque un client s'attend seulement à faire des transferts FTP bidirectionnels, il DEVRAIT produire cette commande aussitôt que possible. Si un client trouve ultérieurement qu'il doit faire un transfert FTP à trois voies après avoir produit une commande EPSV ALL, une nouvelle session FTP DOIT être lancée.

5. Questions de sécurité

Les auteurs estiment que ces changements à FTP n'introduisent pas de nouveaux problèmes de sécurité. Une publication connexe [RFC2577] est une discussion plus générale des questions de sécurité pour FTP et des techniques pour réduire ces problèmes de sécurité.

6. Conclusions

Les extensions spécifiées dans le présent papier permettront à FTP de fonctionner sur divers protocoles réseau.

Références

[RFC2577] M. Allman, S. Ostermann, "Considérations sur la sécurité de FTP", mai 1999. *(Information)*

[RFC1579] S. Bellovin, "Pare-feu FTP facile", février 1994. *(Information)*

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997.

[RFC1883] S. Deering, R. Hinden, "Spécification du protocole Internet, version 6 (IPv6)", décembre 1995. *(Obsolète, voir* RFC2460) *(P.S.)*

[RFC2373] R. Hinden, S. Deering, "Architecture d'adressage IP version 6", juillet 1998. (*Obsolète, voir* RFC4291) *(PS)*

[RFC1639] D. Piscitello, "Fonctionnement de FTP sur les gros enregistrements d'adresse (FOOBAR)", juin 1994. *(Exp.)*

[RFC0791] J. Postel, éd., "Protocole Internet - Spécification du protocole du programme Internet", STD 5, septembre 1981.

[RFC0793] J. Postel (éd.), "Protocole de commande de transmission – Spécification du protocole du programme Internet DARPA", STD 7, septembre 1981.

[RFC0959] J. Postel et J. Reynolds, "Protocole de transfert de fichiers (FTP)", STD 9, octobre 1985. *(MàJ par* RFC7151*)*

[RFC1700] J. Reynolds et J. Postel, "Numéros alloués", STD 2, octobre 1994. *(Historique, voir www.iana.org))*

Adresse des auteurs

Mark Allman
NASA Lewis Research Center/Sterling Software
21000 Brookpark Rd. MS 54-2
Cleveland, OH 44135
téléphone : (216) 433-6586
mél : mallman@lerc.nasa.gov
http://gigahertz.lerc.nasa.gov/~mallman/

Shawn Ostermann
School of Electrical Engineering and Computer Science
Ohio University
416 Morton Hall
Athens, OH 45701
téléphone : (740) 593-1234
mél : ostermann@cs.ohiou.edu

Craig Metz
The Inner Net
Box 10314-1954
Blacksburg, VA 24062-0314
téléphone : (DSN) 754-8590
mél : cmetz@inner.net

Déclaration complète de droits de reproduction