

Groupe de travail Réseau  
**Request for Comments : 743**  
 NIC : 42758

K. Harrenstien, SRI-KL  
 30 décembre 1977  
 Traduction Claude Brière de L'Isle

## Extension FTP : XRSQ/XRCP

La présente RFC décrit une extension à FTP qui permet à l'utilisateur d'un serveur FTP ITS (c'est-à-dire, sur MIT-AI/ML/MC/DMS)) d'envoyer par messagerie le texte d'un message à plusieurs receveurs simultanément ; une telle transmission de message est bien plus efficace que la pratique courante d'envoi du texte pour chaque destinataire supplémentaire sur un site.

Dans cette extension, il y a deux façons de base pour envoyer un seul texte à plusieurs receveurs. Dans l'une, tous les receveurs sont d'abord spécifiés, et ensuite, le texte est envoyé ; dans l'autre, l'ordre est inversé et le texte est envoyé en premier, suivi par les destinataires. Les deux schémas sont nécessaires parce que aucun d'eux n'est optimal par lui-même pour tous les systèmes, comme on va l'expliquer plus loin. Pour choisir un schéma particulier, on utilise la commande XRSQ ; pour spécifier les destinataires après le choix d'un schéma, les commandes XRCP sont données, et pour fournir le texte, les commandes usuelles MAIL ou MLFL s'appliquent.

### Choix du schéma : XRSQ

XRSQ est le moyen par lequel un programme d'utilisateur peut vérifier la mise en œuvre de XRSQ/XRCP, choisir un schéma particulier, rétablir son état à partir de là, et même faire une négociation rudimentaire. Son format est comme celui de la commande TYPE, comme suit :

```
XRSQ [<SP> <schéma>] <CRLF>
```

<schéma> = un seul caractère. Ceux qui sont définis sont les suivants :

R	Receveurs d'abord. Si ce n'est pas mis en œuvre, ce doit être T.
T	Texte d'abord. Si ce n'est pas mis en œuvre, ce doit être R.
?	Demande de préférence. Doit toujours être mis en œuvre.

Aucun argument signifie le "choix" d'aucun des schémas (le schéma par défaut).

Réponses :

200	OK, on va utiliser le schéma spécifié.
215	<schéma> C'est le schéma préféré.
501	Je comprend XRSQ mais ne peut utiliser ce schéma.
5xx	Commande non reconnue ou non mise en œuvre.

Voir à l'Appendice A les choix de codes de réponse.

Trois aspects de XRSQ doivent être soulignés ici. Le premier est qu'un XRSQ sans argument doit toujours retourner une réponse 200 et restaurer l'état par défaut de "Pas de schéma choisi". Toute autre réponse implique que XRSQ et donc XRCP ne sont pas compris ou ne peuvent pas être effectués correctement.

Le second est que l'utilisation de "?" comme <schéma> demande au serveur FTP de retourner une réponse 215 dans laquelle le serveur spécifie un schéma "préféré". Le format de cette réponse est simple :

```
215 <SP> <schéma> [<SP> <texte arbitraire>] <CRLF>
```

Toute autre réponse (par exemple, 4xx ou 5xx) implique que XRSQ et XRCP ne sont pas mis en œuvre, parce que "?" doit toujours être mis en œuvre si il y a XRSQ.

La troisième chose importante au sujet de XRSQ est qu'il a toujours l'effet collatéral de remettre tous les schémas à leur état initial. Ce rétablissement doit être fait quelle que soit la réponse – 200, 215, ou 501. Les actions nécessaires pour un rétablissement seront expliquées lors de l'exposé sur la façon dont fonctionne en fait chaque schéma.

### Spécification du texte de message : MAIL/MLFL

Sans considération du schéma qui a été choisi (s'il en est un) un MAIL ou MLFL avec un argument non nul va se comporter exactement comme précédemment ; la présente extension n'a aucun effet sur eux. Cependant, de telles

commandes MAIL/MLFL normales n'ont pas les mêmes effets collatéraux que XRSQ ; elles "rétablissent" le schéma en cours à son état initial.

C'est seulement lorsque l'argument est nul (par exemple, MAIL<CRLF> ou MLFL<CRLF>) que le schéma particulier utilisé est important, parce que plutôt que de produire une erreur (comme le font actuellement la plupart des serveurs) le serveur va accepter le texte du message pour cette spécifications "nulle" ; ce qui en résulte va dépendre du schéma en cours, et sera décrit dans "Mécanisme du schéma".

### Spécification du receveur : XRCP

Afin de spécifier les noms des receveurs et de recevoir des accusés de réception (ou de refus) pour chaque nom, la nouvelle commande suivante est aussi définie :

XRCP <SP> <Nom du receveur> <CRLF>

Réponse pour Pas de schéma :

507 Pas de schéma encore spécifié ; utiliser XRSQ.

Les réponses pour le schéma T sont identiques à celles pour MAIL/MLFL.

Réponses pour le schéma R (receveurs d'abord) :

200 OK, nom mémorisé.

440 Tableau de receveurs plein, ce nom n'est pas mémorisé.

450 Nom de receveur rejeté. (Permanent !)

520 Nom de receveur rejeté.

4xx Erreur temporaire, réessayer ce nom plus tard.

5xx Erreur permanente, rapport à l'expéditeur.

Voir à l'Appendice A les détails sur le choix des codes de réponse.

Note : L'utilisation de cette commande est une erreur si aucun schéma n'a été encore choisi ; un <schéma> XRSQ doit avoir été donné si XRCP doit être utilisé.

### Mécanisme du schéma : XRSQ R (Receveurs d'abord)

Dans le schéma Receveurs d'abord, XRCP est utilisé pour spécifier les noms que mémorise le serveur FTP dans une liste ou un tableau. Normalement, la réponse pour chaque XRCP va être soit 200 pour l'acceptation, soit un code 4xx/5xx pour un rejet ; le code 450 et tous les codes 5xx sont des rejets permanents (par exemple, usager inconnu) qui devraient faire l'objet d'un rapport à l'expéditeur humain, tandis que les codes 4xx notent en général une erreur temporaire qui peut être rectifiée ultérieurement. Aucune des réponses 4xx/5xx n'a d'impact sur les commandes XRCP précédentes ou suivantes, excepté pour 440 qui indique qu'aucun autre XRCP ne va réussir sauf si un message est envoyé aux destinataires déjà mémorisés ou si un rétablissement est effectué.

L'envoi d'un texte de message aux destinataires mémorisés est fait en passant une commande MAIL ou MLFL sans argument ; c'est-à-dire, juste MAIL<CRLF> ou MLFL<CRLF>. La transmission du texte du message est exactement la même que pour un MAIL/MLFL normal ; cependant, un accusé de réception positif à la fin de la transmission signifie que le message a été envoyé à TOUS les destinataires qui ont été collectés avec XRCP, et un code d'échec signifie qu'on devrait considérer que l'échec concerne TOUS ces destinataires spécifiés. Cela s'applique sans considération du code d'erreur réel ; et que la réponse signifie un succès ou un échec, tous les noms de destinataires mémorisés sont purgés et oubliés - en d'autres termes, les choses sont rétablies dans leur état initial. Cette purge de la liste des noms de destinataires doit aussi être faite comme effet collatéral du "rétablissement" de toute utilisation de XRSQ.

Une réponse 440 à un XRCP peut donc être traitée en utilisant un MAIL/MLFL pour spécifier le message pour les destinataires actuellement mémorisés, et ensuite envoyer d'autres XRCP et un autre MAIL/MLFL, autant de fois que nécessaire ; par exemple, si un serveur n'a de place que pour 10 noms, il va en résulter qu'un message à 50 destinataires sera envoyé 5 fois, à 10 destinataires différents à chaque fois.

Si un usager tente de spécifier un texte de message (MAIL/MLFL sans argument) avant l'envoi d'un XRCP de réussite, cela devrait être traité exactement comme le serait un MAIL/MLFL "normal" avec un destinataire nul ; la plupart des serveurs vont retourner une erreur d'un type quelconque, comme "450 Receveur nul".

Voir à l'Appendice B un exemple qui utilise XRSQ R.

**Mécanisme du schéma : XRSQ T (Texte en premier)**

Dans le schéma avec le texte en premier, MAIL/MLFL sans argument est utilisé pour spécifier un texte de message, que le serveur mémorise à distance. Les XRCP successifs sont alors traités comme si ils étaient des commandes MAIL/MLFL, sauf qu'aucune des manipulations de transfert du texte n'est effectuée ; le texte mémorisé du message est envoyé au destinataire spécifié, et un code de réponse est retourné identique à celui qu'invoquerait un MAIL/MLFL réel. (Noter que TOUT code 2xx indique le succès.)

Le texte du message mémorisé n'est pas oublié jusqu'au prochain MAIL/MLFL ou XRSQ, qui va soit le remplacer par un nouveau texte ou le purger entièrement. Toute utilisation de XRSQ va réinitialiser ce schéma en purgeant le texte mémorisé, comme le fera toute utilisation de MAIL/MLFL avec un argument non nul.

Si un XRCP est vu avant qu'un texte de message ait été mémorisé, l'utilisateur essaye d'envoyer un message vide ; certains serveurs peuvent le permettre, d'autres vont retourner un code d'erreur.

Voir à l'Appendice C un exemple qui utilise XRSQ T.

**Pourquoi deux schémas ?**

Parce qu'aucun n'est par lui-même optimal pour tous les systèmes. XRSQ R permet plutôt un envoi de message "en vrac", parce que tout est sauvegardé et ensuite envoyé simultanément ; cela est très utile pour les systèmes comme ITS où le serveur FTP n'écrit pas lui-même le message directement, mais le fait traiter par un démon (robot) de messagerie central de grande puissance ; plus il y a d'informations (par exemple, les destinataires) associées à un seul "relais", plus la messagerie peut être livrée efficacement.

À l'opposé, XRSQ T est adapté aux serveurs FTP qui veulent livrer directement les messages, de façon incrémentaire un par un. De cette façon ils peuvent retourner un code de réponse individuel de succès/échec pour chaque destinataire qui peut dépendre de facteurs variables du système de fichier tels qu'un dépassement d'allocation de disque, de conflits d'accès à la boîte aux lettres, et ainsi de suite ; si il essayaient d'émuler la messagerie en vrac de XRSQ R, ils auraient à s'assurer qu'une réponse de succès au MAIL/MLFL signifie bien qu'il a été délivré à TOUS les destinataires spécifiés - et pas seulement à certains.

**Notes diverses :**

- \* Comme ceci est après tout une extension au protocole FTP, on doit être prêt à trouver des sites qui ne reconnaissent ni XRSQ ni XRCP. "XRSQ" et "XRSQ ?" sont explicitement conçus comme des vérifications pour voir si l'un ou l'autre schéma est mis en œuvre ; XRCP ne l'est pas, et un retour d'échec de la variété "non mis en œuvre" pourrait être confondu avec "aucun schéma n'a encore été choisi", ou même avec "Destinataire inconnu". Pour être en sécurité, mieux vaut utiliser XRSQ !
- \* Il n'y a aucun moyen pour indiquer dans une réponse positive à "XRSQ ?" que le "schéma" préféré pour un serveur est celui de l'état par défaut ; c'est-à-dire, aucun des schémas multi-destinataires. La raison en est que dans ce cas, il serait sans objet de mettre en œuvre XRSQ/XRCP, et la réponse serait donc négative.
- \* Une des raisons pour lesquelles, avec cette extension multi-destinataires, l'utilisation de MAIL/MLFL est restreinte aux arguments nuls est l'ambiguïté qui résulterait si un argument non nul était permis ; par exemple, si XRSQ R était activé et si des XRCP avaient été donnés, et qu'on faisait un MAIL FOO<CRLF>, il n'y aurait aucun moyen de distinguer une réponse d'échec pour la boîte aux lettres "FOO" d'un échec général pour tous les destinataires spécifiés. Une situation similaire existe pour XRSQ T ; il n'apparaîtrait pas clairement si le texte a été mémorisé et si la boîte aux lettres a échoué, ou l'inverse, ou les deux.
- \* Les "réinitialisations" sont faites par tous les XRSQ et les MAIL/MLFL "normaux" pour éviter les confusions et une mise en œuvre trop compliquée. La commande XRSQ implique un changement, ou une incertitude, d'état, et cette dernière commande devrait autrement utiliser un mécanisme indépendant pour éviter de mettre à mal les bases de données (par exemple, la zone de stockage du texte des messages) utilisées par les schémas T/R. Cependant, une fois qu'un schéma est choisi, il reste "en effet" tout comme un "TYPE A" ou "BYTE 8" reste sélectionné. La façon recommandée pour faire une réinitialisation, sans changer la sélection courante, est avec "XRSQ ?". Se souvenir que "XRSQ" seul revient à l'état pas de schéma.
- \* Il est permis d'introduire d'autres commandes FTP parmi les séquences XRSQ/XRCP/MAIL.

## Appendice A Codes de réponse FTP

Le choix de codes de réponse appropriés pour des commandes nouvelles ou expérimentales est difficile parce qu'il y a eu trois ensembles "officiels" possibles sur lesquels on peut s'appuyer, et il n'apparaît pas clairement lequel peut être utilisé sur un certain site ; ce sont le (1) vieux FTP, (2) nouveau FTP, (3) nouveau FTP révisé. Dans mon choix des allocations de code, j'ai pour la plus grande part ignoré ceux-ci et ai utilisé la RFC 691, "Encore un essai sur le protocole FTP", par Brian Harvey. La raison pour ce faire est la simple observation est que je n'ai jamais entendu parler d'un site qui mette en œuvre le "nouveau FTP", et la RFC 691 incorpore beaucoup de la logique du code de réponse "nouveau FTP" dans le cadre du "vieux FTP". Le seul conflit aigu est traité en permettant que 450 ait sa "vieille" signification, équivalente à 520 – échec permanent. Noter que lorsque on vérifie pour savoir si un site comprend une commande FTP, une réponse de 5xx (précisément, 500) va généralement indiquer, pour tous les ensembles de codes, que la commande n'est pas reconnue.

En passant, je recommande la lecture de la RFC 691 à ceux qui développent des mises en œuvre de FTP ; peut être que si suffisamment de gens s'y mettent, on pourra sortir de ce bourbier.

## Appendice B Exemple de XRSQ R

Exemple de XRSQ R (Receveurs d'abord)

Voici un exemple de la façon dont est utilisé XRSQ R ; l'utilisateur doit d'abord établir que le serveur met bien en œuvre XRSQ :

```
U: XRSQ
S: 200 OK, pas de schéma choisi.
```

Un XRSQ avec un argument nul retourne toujours un 200 si il est mis en œuvre, choisissant le "schéma" de nul, c'est-à-dire, aucun d'entre eux. Si XRSQ n'est pas mis en œuvre, un code de 4xx ou 5xx serait retourné.

```
U: XRSQ R
S: 200 OK, utilisation de ce schéma
```

Tout va bien ; maintenant les destinataires peuvent être spécifiés.

```
U: XRCP Foo
S: 200 OK
```

```
U: XRCP Raboof
S: 520 Destinataire inconnu.
```

```
U: XRCP bar
S: 200 OK
```

Bien, deux sur trois ne sont pas mauvais. Noter que le refus de "Raboof" n'a pas d'effet sur la mémorisation de "Foo" ou "bar". Il faut maintenant fournir le texte du message, en donnant un MAIL ou MLFL sans argument :

```
U: MAIL
S: 350 Taper le message, terminé par <CRLF>.<CRLF>
U: Bla bla bla bla....etc. etc. etc.
U: .
S: 256 Message envoyé.
```

Le texte a maintenant été envoyé aux deux destinataires "Foo" et "bar".

## Appendice C Exemple de XRSQ T

Exemple de XRSQ T (Texte d'abord)

En utilisant le même message que dans l'exemple précédent :

U: XRSQ ?

S: 215 T Texte en premier, s'il vous plaît.

XRSQ est, bien sûr, mis en œuvre, et le serveur dit qu'il préfère "T", mais cela ne doit pas empêcher l'utilisateur d'essayer quelque chose d'autre :

U: XRSQ R

S: 501 Désolé, je ne peux réellement pas faire cela.

Tant pis. Il est possible qu'il aurait pu comprendre aussi "R", mais en général, il vaut mieux utiliser le schéma "préféré", car le serveur sait ce qui est le plus efficace pour son site particulier. De toutes façons :

U: XRSQ T

S: 200 OK, utilisons ce schéma.

Le schéma "T" est maintenant choisi, et le texte doit être envoyé :

U: MAIL

S: 350 Taper le message, terminé par <CRLF>.<CRLF>

U: Bla bla bla bla....etc. etc. etc.

U: .

S: 256 Message mémorisé.

Maintenant les destinataires peuvent être spécifiés :

U: XRCP Foo

S: 256 Message mémorisé envoyé.

U: XRCP Raboof

S: 520 Destinataire inconnu.

U: XRCP bar

S: 256 Message mémorisé envoyé.

Là encore, le texte a été maintenant envoyé à "Foo" et à "bar", et il reste encore mémorisé. Un nouveau message peut être envoyé avec une autre séquence MAIL/XRCP..., mais le fastidieux ou le paranoïaque pourrait choisir de faire :

U: XRSQ ?

S: 215 T Texte en premier, s'il vous plaît.

Qui réinitialise les choses sans altérer le schéma effectif.