

Groupe de travail Réseau
Request for Comments : 3030
 RFC rendue obsolète : 1830
 Catégorie : En cours de normalisation

G. Vaudreuil, Lucent Technologies
 décembre 2000
 Traduction Claude Brière de L'Isle

Extensions de service à SMTP pour la transmission de grands messages MIME binaires

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 (2000). Tous droits réservés.

Résumé

Le présent mémoire définit deux extensions au protocole simple de transfert de messagerie (SMTP, *Simple Mail Transfer Protocol*). La première extension permet aux clients et serveurs SMTP de négocier l'utilisation d'une solution de remplacement à la commande DATA, appelée "BDAT", pour envoyer efficacement de gros messages avec les extensions multi objets de messagerie Internet (MIME, *Multipurpose Internet Mail Extensions*). La seconde extension tire parti de la commande BDAT pour permettre l'envoi négocié de messages MIME qui emploient le codage de transfert binaire. Le présent document est destiné à mettre à jour la [RFC1830] et la rend obsolète.

Résumé du groupe de travail

Le présent protocole n'est pas le produit d'un groupe de travail de l'IETF, cependant, la spécification a résulté de discussions au sein du groupe de travail ESMTP. Le protocole résultant, documenté dans la [RFC1830] a été classé comme expérimental au moment de sa publication du fait de questions sur la robustesse du codage/transfert de contenu binaire déployé dans les mises en œuvre MIME existantes alors. Comme MIME a mûri et que d'autres utilisations du codage/transfert de contenu binaire ont été déployées, ces soucis ont été apaisés. Avec le présent document, l'ESMTP binaire est destiné à devenir une norme.

Conventions pour le document

Les mots clés "DOIT", "NE DOIT PAS", "EXIGÉ", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RRECOMMANDE", "PEUT", et "FACULTATIF" dans ce document sont à interpréter comme défini dans la [RFC2119].

Table des matières

1. Généralités.....	1
2. Cadre des extensions pour grands messages.....	2
3. Cadre pour l'extension de service binaire.....	3
4. Exemples.....	5
4.1 Tronçonnage simple.....	5
4.2 Traitement en parallèle de BINARYMIME.....	5
5. Considérations pour la sécurité.....	5
6. Références.....	6
7. Adresse de l'auteur.....	6
Appendice A Changements par rapport à la RFC 1830.....	6
Déclaration complète de droits de reproduction.....	6

1. Généralités

Les extensions MIME au format de message Internet pourvoient à la transmission de nombreuses sortes de données qui n'étaient précédemment pas prises en charge par la messagerie Internet. Anticipant le besoin de transporter plus efficacement les nouveaux supports, le protocole SMTP a été étendu pour fournir le transport de nouveaux types de messages. La [RFC1652] définit une de ces extensions pour la transmission de messages MIME de 8 bits non codés. Cette

extension de service permet au receveur SMTP de déclarer la prise en charge de parties de corps de 8 bits et à l'envoyeur de demander la transmission en 8 bits d'un message particulier.

Un résultat attendu de l'utilisation de MIME est qu'on s'attend à ce que le système de messagerie de l'Internet transporte de très gros messages. Dans de telles transactions, il y a un désir, fondé sur la performance, d'éliminer l'exigence que le message soit examiné à la recherche des séquences "CR LF . CR LF" à l'envoi et à la réception pour détecter la fin du message.

Indépendamment du besoin d'envoyer de grands messages, la messagerie Internet est de plus en plus multi supports. Il y a un besoin d'éviter la redondance du base64 et du codage quoted-printable des objets binaires envoyés en utilisant le format de message MIME sur SMTP entre des hôtes qui acceptent le traitement de message binaire.

Le présent mémoire utilise le mécanisme défini dans la [RFC1869] pour définir deux extensions au service SMTP par lesquelles un serveur SMTP ("receveur SMTP") peut déclarer la prise en charge du mode de transmission tronçonné de message et la prise en charge de la réception de messages binaires, que le client SMTP ("envoyeur SMTP") est alors libre d'utiliser.

2. Cadre des extensions pour grands messages

L'extension de service suivante est définie :

- 1) Le nom de l'extension de service de tronçonnage des données est "CHUNKING".
- 2) La valeur du mot clé EHLO associée à cette extension est "CHUNKING".
- 3) Un nouveau verbe SMTP, BDAT, est défini comme solution de remplacement à la commande "DATA" de la [RFC821]. Le verbe BDAT prend deux arguments. Le premier argument indique la longueur, en octets, du tronçon de données binaires. Le second argument facultatif indique que le tronçon de données est le dernier.

```
bdat-cmd ::= "BDAT" SP taille de tronçon [ SP marqueur de fin ] CR LF
taille de tronçon ::= 1*CHIFFRE
marqueur de fin ::= "LAST"
```

- 4) Cette extension peut être utilisée pour la soumission de message SMTP [RFC2476].
- 5) Les serveurs qui offrent l'extension BDAT DOIVENT continuer de prendre en charge la commande SMTP DATA normale. Les clients sont libres d'utiliser DATA pour transférer des données correctement codées aux serveurs qui prennent en charge l'extension CHUNKING si ils le souhaitent.

L'extension de service CHUNKING permet l'utilisation de BDAT comme solution de remplacement à la commande DATA. Cette extension peut être utilisée pour tout message, qu'il soit 7 bits, 8BITMIME ou BINARYMIME.

Lorsque un envoyeur SMTP souhaite envoyer (en utilisant la commande MAIL) un grand message en utilisant l'extension CHUNKING, il commence par produire la commande EHLO au receveur SMTP. Si le receveur SMTP répond avec le code 250 à la commande EHLO et si la réponse comporte le mot clé EHLO avec la valeur CHUNKING, le receveur SMTP indique alors qu'il prend en charge la commande BDAT et qu'il va accepter l'envoi de messages en tronçons.

Après que toutes les réponses MAIL et RCPT ont été collectées et traitées, le message est envoyé en utilisant une série de commandes BDAT. La commande BDAT prend un argument obligatoire, la longueur exacte du segment de données en octets. Les données du message sont envoyées immédiatement après le <CR> <LF> de queue de la ligne de commande BDAT. Une fois que le receveur SMTP a reçu le nombre d'octets spécifié, il va retourner un code de réponse 250.

Le paramètre facultatif LAST sur la commande BDAT indique que c'est le dernier tronçon des données du message à être envoyé. La dernière commande BDAT PEUT avoir un compte d'octets de zéro qui indique qu'il n'y a pas de données supplémentaires à envoyer. Toute commande BDAT envoyée après le BDAT LAST est illégale et DOIT recevoir un code de réponse 503 "Mauvaise séquence de commandes". L'état résultant de cette erreur est indéterminé. Une commande RSET DOIT être envoyée pour libérer la transaction avant de continuer.

Une réponse 250 DOIT être envoyée à chaque bloc de données BDAT réussi au sein d'une transaction de messagerie. Si une défaillance survient après réception d'une commande BDAT, le receveur SMTP DOIT l'accepter et éliminer le message de données associé avant d'envoyer le code 5XX ou 4XX approprié. Si un code 5XX ou 4XX est reçu par

l'expéditeur SMTP en réponse à un tronçon BDAT, la transaction devrait être considérée comme un échec et l'expéditeur SMTP NE DOIT PAS envoyer de segment BDAT supplémentaire. Si le receveur SMTP a déclaré la prise en charge du traitement en parallèle de commandes de la [RFC2920], le receveur SMTP DOIT être prêt à l'accepter et à éliminer les tronçons BDAT supplémentaires qui sont déjà en cours de traitement en parallèle (*pipeline*) après le BDAT défaillant.

Note : Une erreur chez le receveur SMTP telle qu'un disque plein ou une fermeture imminente ne peut être rapportée qu'après que le segment BDAT a été reçu. Il est donc important de choisir une taille raisonnable de tronçon en fonction de la bande passante attendue de bout en bout.

Note : Comme le receveur SMTP n'accuse pas réception de la commande BDAT avant l'envoi des données du message, il est important de n'envoyer le BDAT qu'aux systèmes qui ont déclaré leur capacité à accepter les commandes BDAT. L'envoi illégal d'une commande BDAT et des données de message associées à un système qui n'a pas la capacité CHUNKING va résulter en l'analyse par le receveur SMTP des données de message associées comme si il y avait une ligne de commande ESMTP, potentiellement très longue, contenant des données binaires.

L'état résultant de l'échec d'une commande BDAT est indéterminé. Une commande RSET DOIT être produite pour purger la transaction avant que de nouvelles commandes puissent être envoyées. La commande RSET, quand elle est produite après le premier BDAT et avant le BDAT LAST, purge tous les segments envoyés durant cette transaction et redémarre la session.

Les commandes DATA et BDAT ne peuvent pas être utilisées dans la même transaction. Si une déclaration DATA est produite après une BDAT pour la transaction en cours, une réponse 503 "Mauvaise séquence de commandes" DOIT être produite. L'état résultant de cette erreur est indéterminé. Une commande RSET DOIT être envoyée pour purger la transaction avant de continuer. Il n'est pas interdit d'utiliser DATA et BDAT dans la même session, pour autant qu'elles ne soient pas mêlées dans la même transaction.

La taille de mémorisation locale d'un message peut ne pas refléter précisément la taille réelle du message envoyé du fait des conventions locales de mémorisation. En particulier, les messages de texte envoyés avec la commande BDAT DOIVENT être envoyés dans le format MIME canonique avec les lignes délimitées par un <CR><LF>. Il peut n'être pas possible de convertir en une seule fois le message entier au format canonique. CHUNKING fournit un mécanisme pour convertir le message en forme canonique, compter précisément les octets, et envoyer le message en un seul tronçon à la fois.

Note : Le comptage correct des octets est essentiel. Si l'expéditeur SMTP indique une taille de tronçon supérieure à la taille de tronçon réelle, le receveur SMTP va continuer d'attendre le reste des données ou quand on utilise le direct, va lire la commande suivante comme des données supplémentaires du message. Dans le cas où une portion de la commande précédente a été lue comme des données, l'analyseur va retourner une erreur de syntaxe lorsque la commande inachevée sera lue.

Si l'expéditeur SMTP indique une taille de tronçon plus petite que la taille réelle de tronçon, le receveur SMTP va interpréter le reste des données du message comme une commande invalide. Noter que le reste des données du message peut être en binaire et à ce titre, les analyseurs lexicographiques DOIVENT être prêts à recevoir, traiter, et rejeter des lignes d'octets arbitraires.

3. Cadre pour l'extension de service binaire

L'extension de service suivante est définie :

- 1) Le nom de l'extension de service binaire est "BINARYMIME".
- 2) La valeur de mot clé EHLO associée à cette extension est "BINARYMIME".
- 3) L'extension de service BINARYMIME ne peut être utilisée qu'avec l'extension de service "CHUNKING".
- 4) Aucun paramètre n'est utilisé avec le mot clé BINARYMIME.
- 5) La [RFC1652] définit le paramètre BODY pour la commande MAIL. Cette extension définit une valeur supplémentaire pour le paramètre BODY, "BINARYMIME". La valeur "BINARYMIME" associée à ce paramètre indique que ce message est un message binaire MIME (en stricte conformité avec la [RFC2045]) avec l'envoi d'un contenu d'octets arbitraire. La syntaxe révisée de la valeur est comme suit, en utilisant la notation ABNF de la [RFC822]:

valeur-de-corps ::= "7BIT" / "8BITMIME" / "BINARYMIME"

- 6) Aucun nouveau verbe n'est défini pour l'extension BINARYMIME.
- 7) Cette extension peut être utilisée pour la soumission de message SMTP. [RFC2476]
- 8) La longueur maximum d'une ligne de commande MAIL FROM est augmentée de 16 caractères par l'ajout possible du mot clé et de la valeur BODY=BINARYMIME;

Un expéditeur SMTP peut demander qu'un message MIME binaire soit envoyé sans codage de transport en envoyant un paramètre BODY avec une valeur de "BINARYMIME" avec la commande MAIL. Lorsque le receveur SMTP accepte une commande MAIL avec la valeur de corps BINARYMIME, il accepte de préserver tous les bits dans chaque octet passé en utilisant la commande BDAT. Une fois qu'un receveur SMTP qui prend en charge l'extension de service BINARYMIME a accepté un message contenant du matériel binaire, le receveur SMTP DOIT livrer ou relayer le message d'une façon telle qu'elle préserve tous les bits de chaque octet.

BINARYMIME ne peut pas être utilisé avec la commande DATA. Si une commande DATA est produite après une commande MAIL contenant la valeur-de-corps de "BINARYMIME", une réponse 503 "Mauvaise séquence de commandes" DOIT être envoyée. L'état résultant de cette condition d'erreur est indéterminé et la transaction DOIT être redémarrée avec la commande RSET.

Il est particulièrement important de s'assurer, lorsque on utilise BINARYMIME, que le message MIME lui-même est correctement formé. En particulier, il est essentiel que le texte soit codé de façon canonique avec chaque ligne bien terminée par un <CR><LF>. Toute transformation de texte en MIME non canonique pour respecter des conventions locales de mémorisation DOIT être inversée avant un envoi comme BINARYMIME. Certains raccourcis fondés sur la ligne vont échouer si ils sont utilisés avec BINARYMIME. Un expéditeur SMTP DOIT utiliser le codage canonique pour un certain type de contenu MIME. En particulier, text/* DOIT être envoyé avec des lignes terminées par des <CR><LF>.

Note : Bien que CR et LF ne représentent pas nécessairement des fins de lignes de texte dans des tronçons BDAT et que l'utilisation de codage de transfert binaire soit admise, la RFC2781 interdit d'utiliser un jeu de caractères UTF-16 au sein du type de support text de niveau supérieur.

La syntaxe de la commande MAIL étendue est identique à celle de la commande MAIL de la [RFC821], sauf que le paramètre et la valeur BODY=BINARYMIME DOIVENT être ajoutés. La syntaxe complète de cette commande étendue est définie dans la [RFC1869].

Si un receveur SMTP n'indique pas la prise en charge du format de message BINARYMIME, l'expéditeur SMTP NE DOIT alors, dans aucune circonstances, envoyer de données binaires.

Si le receveur SMTP ne prend pas en charge BINARYMIME et si le message à envoyer est un objet MIME avec un codage binaire, un expéditeur SMTP a trois options pour transmettre le message. D'abord, si le receveur SMTP prend en charge l'extension de transport MIME 8bit de la [RFC1652] et si le contenu est éligible au codage en 8bit, l'expéditeur SMTP peut mettre en œuvre une transformation de passerelle pour convertir le message en MIME codé en 8bit valide. Ensuite, il peut mettre en œuvre une transformation de passerelle pour convertir le message en MIME codé en 7bit valide. Enfin, il peut traiter cela comme une erreur permanente et gérer cela de la façon usuelle pour les livraisons défectueuses. Les spécificités des codages de transfert de contenu MIME, y compris les transformations de MIME binaire en MIME à 8bit ou MIME à 7bit ne sont pas décrites dans la présente RFC ; la conversion est néanmoins soumise aux contraintes suivantes :

1. La conversion DOIT ne causer aucune perte d'information ; les codages de transport MIME DOIVENT être employés comme nécessaire pour garantir que c'est le cas.
2. Le message résultant DOIT être du MIME 7bit ou 8bit valide. En particulier, la transformation NE DOIT PAS résulter en un codage de transfert de contenu en Base-64 ou Quoted-Printable.

Note : Au moment de la rédaction du présent document, il n'existe pas de mécanisme pour convertir un objet MIME binaire en objet MIME 8-bit. Une telle transformation exigera la spécification d'un nouveau codage de transfert de contenu MIME.

Si le message MIME contient un codage de transfert de contenu "Binary" et si le paramètre BODY n'indique pas BINARYMIME, le message DOIT être accepté. Le message DEVRAIT être retourné à l'expéditeur avec une notification de l'état de livraison (DSN, *Delivery State Notification*) appropriée. Le contenu du message PEUT être retourné à l'expéditeur si le contenu incriminé peut être inclus dans une structure légale de DSN. La "réparation" et la transmission du contenu incriminé sortent du domaine d'application du présent document.

4. Exemples

4.1 Tronçonnage simple

Le dialogue simple suivant illustre l'utilisation de l'extension de grand message pour envoyer un bref message pseudo-RFC822 à un receveur en utilisant l'extension CHUNKING :

```
R: <attente de la connexion sur l'accès TCP 25>
S: <connexion ouverte avec le serveur>
R: 220 cnri.reston.va.us SMTP service prêt
S: EHLO ymir.claremont.edu
R: 250-cnri.reston.va.us dit hello
R: 250 CHUNKING
S: MAIL FROM:<Sam@Random.com>
R: 250 <Sam@Random.com> Sender ok
S: RCPT TO:<Susan@Random.com>
R: 250 <Susan@random.com> Receveur ok
S: BDAT 86 LAST
S: To: Susan@random.com<CR><LF>
S: From: Sam@random.com<CR><LF>
S: Subject: Ceci est un message d'essai sans corps<CR><LF>
R: 250 Message OK, 86 octets reçus
S: QUIT
R: 221 Goodbye
```

4.2 Traitement en parallèle de BINARYMIME

Le dialogue suivant illustre l'utilisation de l'extension de grand message pour envoyer un objet BINARYMIME à deux receveurs en utilisant les extensions CHUNKING et PIPELINING :

```
R: <attente de connexion sur l'accès TCP>
S: <connexion ouverte avec le serveur>
R: 220 cnri.reston.va.us SMTP service prêt
S: EHLO ymir.claremont.edu
R: 250-cnri.reston.va.us dit hello
R: 250-PIPELINING
R: 250-BINARYMIME
R: 250 CHUNKING
S: MAIL FROM:<ned@ymir.claremont.edu> BODY=BINARYMIME
S: RCPT TO:<gvaudre@cnri.reston.va.us>
S: RCPT TO:<jstewart@cnri.reston.va.us>
R: 250 <ned@ymir.claremont.edu>... Sender and BINARYMIME ok
R: 250 <gvaudre@cnri.reston.va.us>... Recipient ok
R: 250 <jstewart@cnri.reston.va.us>... Recipient ok
S: BDAT 100000
S: (D'abord 10000 octets de données de message MIME canonique)
S: BDAT 324
S: (Restent 324 octets de données de message MIME canonique)
S: BDAT 0 LAST
R: 250 100000 octets reçus
R: 250 324 octets reçus
R: 250 Message OK, 100324 octets reçus
S: QUIT
R: 221 Goodbye
```

5. Considérations pour la sécurité

Il n'est pas connu que ces extensions présente de risques supplémentaires pour la sécurité par rapport à ceux qui sont déjà endémiques dans la messagerie électronique et sont présents dans les mises en œuvre pleinement conformes à la [RFC821], ou autrement rendus possibles par la [RFC2045].

6. Références

- [RFC0821] J. Postel, "Protocole simple de [transfert de messagerie](#)", STD 10, août 1982.
- [RFC0822] D. Crocker, "Norme pour le [format des messages de texte](#) de l'ARPA-Internet", STD 11, août 1982. (*Obsolète, remplacée par la RFC5322*)
- [RFC1652] J. Klensin et autres, "[Extensions de service SMTP](#) pour transport MIME sur 8 bits", juillet 1994. (*Remplacée par RFC6152*) (D.S.)
- [RFC1830] G. Vaudreuil, "Extensions de service SMTP pour la transmission de gros messages MIME binaires", août 1995. (*Expérimentale*) (*Obsolète, voir RFC3030*)
- [RFC1869] J. Klensin, N. Freed, M. Rose, E. Stefferud, D. Crocker, "Extensions de service à SMTP", novembre 1995. (*Obsolète, voir RFC5321, STD0010*)
- [RFC2045] N. Freed et N. Borenstein, "[Extensions de messagerie Internet](#) multi-objets (MIME) Partie 1 : Format des corps de message Internet", novembre 1996. (*D. S., MàJ par 2184, 2231, 5335.*)
- [RFC2119] S. Bradner, "[Mots clés à utiliser](#) dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997.
- [RFC2476] R. Gellens, J. Klensin, "Soumission de message", décembre 1998. (*Obsolète, voir RFC4409*) (P.S.)
- [RFC2920] N. Freed, "Extension de service SMTP pour le [traitement de commandes en parallèle](#)", septembre 2000. (*STD0060*)

7. Adresse de l'auteur

Gregory M. Vaudreuil
Lucent Technologies
17080 Dallas Parkway
Dallas, TX 75248-1905
USA
téléphone/fax : 972-733-2722
mél : GregV@ieee.org

Appendice A Changements par rapport à la RFC 1830

De nombreux changements rédactionnels y compris le couplet obligé sur la propriété intellectuelle et des informations révisées pour le contact avec les auteurs.

Correction de l'exemple de tronçonnage simple pour utiliser le nombre d'octets correct. Mise à jour de l'exemple de fonctionnement en parallèle pour illustrer l'utilisation de la construction BDAT 0 LAST.

Déclaration complète de droits de reproduction

Copyright (c) The Internet Society (2000). Tous droits réservés.

Le présent document et ses traductions peuvent être copiés et fournis aux tiers, et les travaux dérivés qui les commentent ou les expliquent ou aident à leur mise en œuvre peuvent être préparés, copiés, publiés et distribués, en tout ou partie, sans

restriction d'aucune sorte, pourvu que la déclaration de copyright ci-dessus et le présent et paragraphe soient inclus dans toutes telles copies et travaux dérivés. Cependant, le présent document lui-même ne peut être modifié d'aucune façon, en particulier en retirant la notice de droits de reproduction ou les références à la Internet Society ou aux autres organisations Internet, excepté autant qu'il est nécessaire pour le besoin du développement des normes Internet, auquel cas les procédures de droits de reproduction définies dans les procédures des normes Internet doivent être suivies, ou pour les besoins de la traduction dans d'autres langues que l'anglais.

Les permissions limitées accordées ci-dessus sont perpétuelles et ne seront pas révoquées par la Internet Society, ses successeurs ou ayant droits.

Le présent document et les informations y contenues sont fournies sur une base "EN L'ÉTAT" et le contributeur, l'organisation qu'il ou elle représente ou qui le/la finance (s'il en est), la INTERNET SOCIETY et la INTERNET ENGINEERING TASK FORCE déclinent toutes garanties, exprimées ou implicites, y compris mais non limitées à toute garantie que l'utilisation des informations ci encloses ne violent aucun droit ou aucune garantie implicite de commercialisation ou d'aptitude à un objet particulier.

Remerciement

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