Groupe de travail Réseau
Request for Comments : 4236
Catégorie : Sur la voie de la normalisation
Traduction Claude Brière de L'Isle

A. Rousskov,The Measurement Factory
M. Stecher, CyberGuard Corporation

novembre 2005

Adaptation HTTP avec services marginaux à connexion libre (OPES)

Résumé
Le cadre de services marginaux à connexion libre (OPES, *Open Pluggable Edge Services*) documente plusieurs mécanismes qui ignorent l'application comme le traçage OPES, l'outrepassement OPES, et le protocole d'invocation OPES. Le présent document étend ces mécanismes génériques pour l'adaptation du protocole de transfert hypertexte (HTTP, *Hypertext Transfer Protocol*). Ensemble, les documents OPES ignorant de l'application et le présent profil HTTP constituent une spécification complète pour l'adaptation HTTP avec OPES.

Table des matières

1. Domaine d'application

Le cadre de services marginaux à connexion libre (OPES, *Open Pluggable Edge Services*) documente plusieurs mécanismes indifférents à l'application comme les processeurs OPES et les communications de point d'extrémité [RFC3897] ou le protocole d'invocation OPES [RFC4037]. Le présent document étend ces mécanismes génériques pour l'adaptation d'un protocole spécifique d'application, HTTP [RFC2616]. Ensemble, les documents OPES qui ignorent l'application et le présent profil HTTP constituent une spécification complète pour l'adaptation HTTP avec OPES.

Les sections principales de ce document spécifient des extensions spécifiques de HTTP pour le mécanisme correspondant ignorant de l'application qui est documenté ailleurs.

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans le BCP 14, [RFC2119].

2 Cartographie des documents OPES

Le présent document appartient à un grand ensemble de spécifications OPES produites par le groupe de travail OPES de l'IETF. La familiarité avec l'approche globale de OPES et des scénarios typiques est souvent essentielle pour essayer de comprendre des documents OPES isolés. La présente section fournit un index des documents OPES pour aider le lecteur à trouver les informations "manquantes".

- Le document sur les cas d'utilisation et les scénarios de déploiement de OPES [RFC3752] décrit un ensemble de services et applications qui sont examinés dans le domaine des OPES et ont été utilisés comme motifs et lignes directrices pour concevoir l'architecture OPES.

- L'architecture OPES et la terminologie commune sont décrites dans "Architecture pour les services marginaux à connexion libre (OPES)" [RFC3835].

- "Exigences de politique, d'autorisation et de mise en œuvre pour OPES" [RFC3838] précise les exigences et les hypothèses sur le cadre de politique, sans spécifier de méthodes concrètes d'autorisation et d'application.

- "Menaces et risques pour la sécurité des OPES" [RFC3837] fournit une analyse de risque pour OPES, sans recommander de solutions spécifiques.

- "Traitement par OPES des considérations de l'IAB" [RFC3914] adresse toutes les considérations de niveau architecture exprimées par le Bureau d'architecture de l'Internet (IAB) de l'IETF quand a été élaboré le mandat du groupe de travail OPES.

- Au cœur de l'architecture OPES sont les processeurs OPES et le serveur d'invocation, deux éléments de réseau qui communiquent ensemble via un protocole d'invocation d'OPES (OCP, *OPES Callout Protocol*). Les exigences pour ce protocole sont discutées dans "Exigences pour les protocoles d'invocation d'OPES" [RFC3836].

- "Cœur du protocole d'invocation d'OPES" [RFC4037] spécifie un protocole qui ignore les applications pour être utilisé pour la communication entre le processeur OPES et un serveur d'invocation.

- "Communications des entités OPES et des points d'extrémité" [RFC3897] spécifie les mécanismes génériques de traçage et d'outrepassement pour OPES.

- Les documents du cœur de OCP et des communications sont indépendants du protocole d'application adapté par les entités OPES. Les mécanismes génériques sont complétés par des profils spécifiques des applications. Le présent document, adaptation HTTP avec OPES, est un tel profil d'application pour HTTP. Il spécifie comment des mécanismes OPES ignorants des applications sont à utiliser et à augmenter afin de prendre en charge l'adaptation des messages HTTP.

- Finalement, "P : langage de traitement de messages" [rules-p] définit un langage pour spécifier quelles adaptations OPES (par exemple, la traduction) doivent être appliquées à quels messages d'application (par exemple, les messages électroniques provenant de bob@exemple.com). Le langage P est destiné à la configuration des applications de mandataires (processeurs OPES).

3. Protocole d'invocation

Cette section documente le profil HTTP pour le cœur du protocole d'invocation de OPES (OCP) [RFC4037]. La familiarité avec le cœur OCP est nécessaire pour comprendre le profil HTTP. Cette section utilise les conventions, la terminologie, et les mécanismes du cœur OCP.

Le processeur OPES communique son désir d'adapter les messages HTTP via un messages d'offre de négociation (NO, *Negotiation Offer*) avec les identifiants de caractéristique spécifiques de HTTP documentés au paragraphe 3.2. Les mécanismes d'optimisation OCP spécifiques de HTTP peuvent être négociés en même temps. Un serveur d'invocation qui prend en charge l'adaptation des messages HTTP a une chance de négocier les parties de message HTTP qui vont participer à l'adaptation, incluant la négociation des parties de demande HTTP comme les métadonnées pour l'adaptation de la réponse HTTP. Les parties de message HTTP négociables sont décrites au paragraphe 3.1.

Le profil HTTP introduit un nouveau paramètre pour le message de début de message d'application (AMS, *Application Message Start*) pour communiquer la longueur de message HTTP connue (les en-têtes HTTP ne portent souvent pas d'informations fiables de longueur, ou pas du tout). Ce paramètre est décrit au paragraphe 3.3. Le paragraphe 3.4 décrit un mécanisme pour faire rapport des parties de message HTTP avec ces messages "Utiliser mes données" (DUM, *Data Use Mine*).

Les paragraphes d'OCP restants documentent divers cas marginaux de surveillance d'OCP comme le traitement des codages de transfert HTTP et les réponses 100 Continue.

3.1 Parties de message d'application

Un message HTTP peut avoir plusieurs parties bien connues : en-têtes, corps, et en queues. Les processeurs HTTP OPES vont probablement avoir des informations sur les parties de message HTTP parce qu'ils doivent isoler et interpréter les en-têtes HTTP et trouver les limites de message HTTP. Les serveurs d'invocation peuvent ne pas se soucier de certaines parties ou peuvent bénéficier de la réutilisation du travail du processeur OPES HTTP sur l'isolement et la catégorisation des parties intéressantes.

Voici la déclaration d'un type am-part (partie de message d'application) qui utilise le mnémonique de déclaration de type d'élément de protocole cœur OCP (PETDM, *Protocol Element Type Declaration Mnemonic*) :

```
am-part:  étend atome ;
am-parts: étend liste de am-part ;
```

Figure 1

Les six atomes "am-part" suivants sont des valeurs valides :

request-header *(en-tête de demande)* : ligne de départ d'un message de demande HTTP, de tous les en-têtes de message de demande, et le séparateur CRLF à la fin des en-têtes HTTP (comparer au paragraphe 4.1 de la [RFC2616]).

request-body *(corps de demande)* : corps de message d'un message de demande HTTP comme défini au paragraphe 4.3 de la [RFC2616] mais sans inclure l'en-queue.

request-trailer *(en-queue de demande)* : en-têtes d'entité de l'en-queue d'un message de demande HTTP en codage de transfert tronqué. Cette partie suit la même syntaxe que l'en-queue défini au paragraphe 3.6.1 de la [RFC2616].

response-header *(en-tête de réponse)* : ligne de début d'un message de réponse HTTP , de tous les en-têtes de message de réponse, et le séparateur CRLF à la fin des en-têtes HTTP (à comparer au paragraphe 4.1 de la [RFC2616]).

response-body *(corps de réponse)* : corps de message d'un message de réponse HTTP comme défini au paragraphe 4.3 de la [RFC2616] mais non inclus les en-queues.

response-trailer *(en-queue de réponse)* : en-têtes d'entité de l'en-queue d'un message de réponse HTTP en codage de transfert tronqué. Cette partie suit la même syntaxe que l'en-queue défini au paragraphe 3.6.1 de la [RFC2616].

3.2 Caractéristiques de profil d'application

Le présent document définit deux profils HTTP pour OCP : profil de demande et profil de réponse. Ces deux profils sont décrits ci-dessous. Chaque profil a un identifiant de caractéristique unique, une liste de parties originales de messages d'application, et une liste de parties adaptées de message d'application :

identifiant de profil : http://www.iana.org/assignments/opes/ocp/http/request

parties originales de demande : request-header, request-body, request-trailer

parties adaptées de demande : request-header, request-body, request-trailer

parties adaptées de réponse : response-header, response-body, response-trailer

identifiant de profil : http://www.iana.org/assignments/opes/ocp/http/response

parties originales de transaction : request-header (aux), request-body (aux), request-trailer (aux), response-header, response-body, response-trailer

parties adaptées de réponse : response-header, response-body, response-trailer

Le profil de demande contient deux variantes de liste de parties adaptées : parties de demande HTTP et parties de réponse HTTP. Les parties marquées d'un suffixe "(aux)" sont les parties auxiliaires qui peuvent seulement être utilisées si elles sont explicitement négociées pour un profil. Voir au pargraphe 3.2.1 les règles spécifiques qui gouvernent la négociation et l'utilisation des am-parts.

La portée d'un profil négocié est la connexion OCP (par défaut) ou le groupe de service spécifié via le paramètre SG.

3.2.1 Parties de profil

Un agent OCP DOIT envoyer les parties de message d'application dans l'ordre impliqué par les listes de parties de profil ci-dessus. Un agent OCP qui reçoit une partie déclassée PEUT terminer la transaction avec une erreur.

Un processeur OPES NE DOIT PAS envoyer de parties qui ne figurent pas comme "originales" sur la liste dans le profil négocié. Un serveur d'invocation NE DOIT PAS envoyer des parties qui ne figurent pas sur la liste comme "adaptées" dans le profil négocié. Un agent OCP qui reçoit une partie non listée DOIT terminer la transaction avec une erreur. La raison informelle de cette dernière exigence est de réduire le nombre de problèmes subtils d'interopérabilité lorsque un agent pense que les parties qu'il envoie sont comprises par l'autre agent quand, en fait, elles ont été ignorées ou sautées parce qu'elles ne sont pas attendues.

Il manque certaines parties à certains messages HTTP. Par exemple, de nombreuses demandes HTTP n'ont pas de corps, et la plupart des messages HTTP n'ont pas d'en-queue. Un agent OCP NE DOIT PAS envoyer (c'est-à-dire, doit sauter) les parties absentes de message d'application.

Un agent OCP DOIT envoyer les parties présentes non auxiliaires et il DOIT envoyer les parties auxiliaires présentes qui ont été négociées via le paramètre Aux-Parts (paragraphe 3.2.3). Les agents OCP NE DOIVENT PAS envoyer de parties auxiliaires qui n'ont pas été négociées via le paramètre Aux-Parts.

Un agent OCP qui reçoit une partie de message en violation des exigences ci-dessus PEUT terminer la transaction correspondante avec une erreur.

Par conception, les parties originales non incluses dans la liste des parties adaptées ne peuvent pas être adaptées. En d'autres termes, un service d'invocation peut seulement adapter les parties dans les listes de parties adaptées même si il peut avoir accès aux autres parties.

Dans le profil de demande, le serveur d'invocation DOIT envoyer soit des parties de demande adaptées, soit des parties de réponse adaptées. Un processeur OPES qui reçoit un flux adapté avec des parties de message d'application provenant des deux listes (en violation de la règle précédente) DOIT terminer la transaction OCP avec une erreur. De façon informelle, le serveur d'invocation envoie des parties de réponse adaptées pour "court-circuiter" la transaction HTTP, forçant le processeur OPES à retourner une réponse HTTP sans transmettre une demande HTTP adaptée. Ce court-circuit est utile pour répondre, par exemple, à une demande HTTP que le service d'invocation définit comme interdite.

Sauf explicitement configuré pour faire autrement, un processeur OPES DOIT offrir toutes les parties originales non auxiliaires dans les messages d'offre de négociation (NO, *Negotiation Offer*). Voir au paragraphe 3.5 la raison de cette règle et des exemples d'effets collatéraux dommageables d'adaptation sélective.

3.2.2 Structure de profil

Une caractéristique de profil d'application HTTP étend la sémantique du type de caractéristique du cœur OCP tout en ajoutant à ce type les paramètres nommés suivants :

- Aux-Parts *(parties auxiliaires)* (paragraphe 3.2.3)

- Pause-At-Body *(pause au corps)* (paragraphe 3.2.4)
- Stop-Receiving-Body *(arrêt de réception du corps)* (paragraphe 3.2.5)
- Preservation-Interest-Body *(intérêt pour la préservation du corps)* (paragraphe 3.2.6)
- Content-Encodings *(codages de contenu)* (paragraphe 3.2.7)

Voici la définition de la structure de caractéristique de profil HTTP en utilisant un PETDM :

```
HTTP-Profile : étend la caractéristique avec {
     [Aux-Parts : am-parts] ;
     [Pause-At-Body : size] ;
     [Stop-Receiving-Body : size] ;
     [Preservation-Interest-Body : size] ;
     [Content-Encoding s: encodings] ;
  };
```

Une structure de profil HTTP peut être utilisée dans les listes de caractéristiques des messages d'offre de négociation (NO) et comme paramètre anonyme d'un message de réponse de négociation (NR, *Negotiation Response*). Tous les paramètres de profil s'appliquent à toutes les transactions OCP au sein d'un profil.

3.2.3 Aux-Parts

Le paramètre Aux-Parts d'un profil de réponse HTTP peut être utilisé pour négocier l'inclusion de parties auxiliaires de message d'application dans le flux de données d'origine. Le paramètre est une liste éventuellement vide de jetons am-part. Un processeur OPES PEUT envoyer un paramètre Aux-Parts pour annoncer la disponibilité des parties auxiliaires de message d'application. Un serveur d'invocation PEUT répondre avec un sous ensemble éventuellement vide des parties dont il a besoin. La réponse du serveur d'invocation définit le sous ensemble des parties auxiliaires de messages dont la négociation a réussi.

Quand il reçoit un message d'offre de négociation (NO) le serveur d'invocation DOIT ignorer toute partie non auxiliaire mentionnée dans le paramètre Aux-Parts. Quand il envoie un message de réponse de négociation (NR) le serveur d'invocation NE DOIT PAS choisir une partie de message d'application qui ne figurait pas explicitement sur la liste de l'offre de négociation. En cas de violation de cette dernière règle, le processeur OPES DOIT terminer la transaction.

Un processeur OPES DOIT envoyer chaque partie auxiliaire négociée au serveur d'invocation, sauf si la partie est absente.

Exemple : Aux-Parts: (request-header,request-body)

3.2.4 Pause-At-Body

Un serveur d'invocation PEUT utiliser le paramètre Pause-At-Body pour demander une pause dans la transmission du message d'application original avant que le flux de données original commence. La valeur du paramètre est de type "offset" *(décalage)*. Le paramètre spécifie le début du suffixe de corps de message d'application non auxiliaire que l'envoyeur n'est temporairement pas intéressé à voir.

```
[en-têtes][ préfixe de corps | suffixe de corps ][en-queue]
<-- ? ---><---décalage  ----><---------- ? ---------->
<-- équiv. décalage DWP ->
```

Quand un processeur OPES reçoit un paramètre Pause-At-Body, il DOIT se comporter comme si il avait reçu un message "Veux une pause des données" (DWP, *Want Data Paused*) avec le org-offset correspondant. Noter que ce dernier décalage est différent du décalage Pause-At-Body et est inconnu tant que la taille des en-têtes du message HTTP n'est pas connue.

Par exemple, si la valeur de Pause-At-Body est zéro, le processeur OPES devrait envoyer un message "Pause dans mes données" (DPM, *Paused My Data*) juste avant d'envoyer le premier message "Utiliser mes données" (DUM, *Data Use Mine*) avec la partie corps de réponse dans le profil de réponse HTTP. Si la valeur de Pause-At-Body est 300, le processeur OPES devrait envoyer un message DPM après la transmission de 300 octets pour cette partie de message d'application.

Exemple : Pause-At-Body: 0

Un serveur d'invocation PEUT utiliser le paramètre Stop-Receiving-Body pour impliquer un comportement de message "Veux arrêter de recevoir des données" (DWSR, *Want Stop Receiving Data*) avant que commence le flux original de

données. La valeur du paramètre est du type "offset". Le paramètre spécifie un décalage dans la partie de corps de message non auxiliaire originale (request-body dans le profil de demande et response-body dans le profil de réponse).

Un service d'invocation PEUT envoyer un paramètre Stop-Receiving-Body avec sa réponse de négociation si il y a un décalage fixe dans le corps de message pour toutes les transactions d'un profil pour lequel un message DWSR aurait été envoyé. Un processeur OPES DOIT se comporter comme si il avait reçu un message DWSR avec le décalage correspondant. Noter que ce dernier décalage est différent du décalage du Stop-Receiving-Body et qu'il n'est pas connu tant que la taille des en-têtes du message HTTP n'est pas connue.

Par exemple, si la valeur de Stop-Receiving-Body est zéro dans un profil de réponse HTTP, le processeur OPES devrait envoyer un message "Fin de message d'application" (AME, *Application Message End*) avec un code de résultat de 206 immédiatement après l'envoi de la partie en-tête de réponse du message et avant de commencer la partie corps de réponse du message.

Exemple : Stop-Receiving-Body: 0

3.2.6 Preservation-Interest-Body

Le paramètre "Preservation-Interest-Body" peut être utilisé pour optimiser la préservation des données au processeur OPES. La valeur du paramètre est du type "size" et désigne une taille de préfixe de la partie originale, non auxiliaire du corps de message (request-body dans le profil de demande HTTP et response-body dans le profil de réponse).

Un service d'invocation PEUT envoyer un paramètre Preservation-Interest-Body avec sa réponse de négociation si il y a un préfixe de taille fixe du corps de message d'application pour lequel un message "Intérêt de préservation des données" (DPI, *Data Preservation Interest*) aurait été envoyé. Un processeur OPES DOIT se comporter comme si il recevait un message DPI avec org-offset zéro et org-size égal à la valeur du paramètre Preservation-Interest-Body.

Par exemple, si la valeur de Preservation-Interest-Body est zéro dans un profil de réponse HTTP, le serveur d'invocation ne doit pas envoyer de message "Utiliser vos données (DUY, *Data Use Yours*) pour la partie corps de réponse ; le processeur OPES peut utiliser cette information pour optimiser son comportement de préservation des données même avant qu'il prenne la décision de préserver les données.

Exemple : Preservation-Interest-Body: 0

3.2.7 Content-Encodings

Un serveur d'invocation PEUT envoyer une liste de Content-Encodings *(codages de contenu)* pour indiquer ses préférences de codage de contenu. Les premiers codages de la liste sont préférés aux autres codages. Un processeur OPES PEUT utiliser tout codage de contenu pour envoyer des messages d'application à un serveur d'invocation.

La liste des codages de contenu préférés n'implique pas une absence de prise en charge des autres codages. Le processeur OPES NE DOIT PAS outrepasser un service juste parce que le codage de contenu réel ne correspond pas aux préférences du service.

Si un agent OCP reçoit un message d'application qu'il ne peut pas traiter à cause d'un codage de contenu spécifique, les règles usuelles de terminaison de transaction s'appliquent.

content-coding : étend atome ;
content-codings : étend liste de content-coding;

Exemple : Content-Encodings: (gzip)

La sémantique de "content-coding" est définie au paragraphe 3.5 de la [RFC2616].

3.2.8 Exemple de négociation de profil

Exemple :

```
P: NO ({"54:http://www.iana.org/assignments/opes/ocp/http/response"
  Aux-Parts: (en-tête-de-demande,corps-de-demande)
  })
  SG: 5
  ;
```

```
S: NR {"54:http://www.iana.org/assignments/opes/ocp/http/response"
  Aux-Parts: (en-tête-de-demande)
  Pause-At-Body: 30
  Preservation-Interest-Body: 0
  Content-Encodings: (gzip)
  }
  SG: 5
  ;
```

Cet exemple montre une offre de négociation faite par un processeur OPES pour un groupe de service (id 5) qui a déjà été créé ; le serveur d'invocation envoie une réponse de négociation adéquate.

Le processeur OPES offre une caractéristique de profil pour les messages de réponse HTTP. À côté des parties standard de message, le processeur OPES est capable d'ajouter l'en-tête et le corps de la demande HTTP originale comme parties auxiliaires du message.

Le serveur d'invocation demande la partie auxiliaire de l'en-tête de demande, mais n'est pas intéressé à recevoir la partie corps de demande.

Le processeur OPES envoie au plus les parties de message suivantes, dans l'ordre spécifié, pour toutes les transactions dans le groupe de service 5 : en-tête-de-demande, en-tête-de-réponse, corps-de-réponse, en-queue-de-réponse. Noter que la partie corps-de-demande n'est pas incluse (parce que c'est une partie auxiliaire qui n'était pas explicitement demandée). Certaines parties de réponse peuvent n'être pas envoyées si elles manquent dans le message original.

Le serveur d'invocation indique par le paramètre Preservation-Interest-Body réglé à zéro qu'il ne va pas envoyer de message DUY. Le processeur OPES peut donc ne rien préserver pour les transactions de ce profil.

En envoyant une valeur de Pause-At-Body de 30, le serveur d'invocation demande une pause des données. Le processeur OPES envoie un message "Pause de mes données" (DPM) immédiatement après l'envoi d'au moins 30 octets de la partie corps-de-réponse. Ensuite, le processeur OPES attend un message "Veux plus de données" (DWM) du service d'invocation.

3.3 Message de début de message d'application

On introduit un nouveau paramètre nommé pour les message "Début de message d'application" (AMS, *Application Message Start*).

AM-EL: size

La valeur de AM-EL est la taille de la partie corps-de-demande dans le profil de demande HTTP, et c'est la taille de la partie corps-de-réponse dans le profil de réponse HTTP, avant qu'un codage de transfert soit appliqué (ou après que tous les codages de transfert ont été retirés). Cette définition est cohérente avec la définition de longueur d'entité HTTP.

Un agent OCP qui connaît la longueur exacte de l'entité message HTTP (voir au paragraphe 7.2.2 "Longueur d'entité" dans la [RFC2616]) au moment où il envoie le message AMS, DEVRAIT annoncer cette longueur en utilisant le paramètre nommé AM-EL d'un message AMS. Si la longueur d'entité exacte n'est pas connue, un agent OCP NE DOIT PAS envoyer de paramètre AM-EL. Relayer la longueur d'entité correcte peut avoir des avantages de performances significatifs pour le receveur, et les mises en œuvre sont fortement encouragées à relayer les longueurs d'entité connues. De même, relayer des longueurs d'entité incorrectes peut avoir des conséquences drastiques pour le receveur, et les mises en œuvre devraient faire très attention quand elles relaient une longueur d'entité.

Un processeur OPES qui reçoit un paramètre AM-EL DEVRAIT utiliser la valeur du paramètre dans un en-tête d'entité HTTP Content-Length *(longueur de contenu)* quand il construit un message HTTP, pourvu qu'un en-tête d'entité Content-Length soit permis pour ce message d'application par HTTP (voir le paragraphe 3.8.1).

3.4 Message DUM

Un nouveau paramètre nommé pour les messages "Utiliser mes données" (DUM, *Data Use Mine*) est introduit.

AM-Part: am-part

Un agent OCP DOIT envoyer un paramètre AM-Part avec chaque message DUM qui fait partie d'une transaction OCP avec un profil HTTP. La valeur du paramètre AM-Part est un seul jeton am-part. Comme l'implique la syntaxe, un message DUM peut seulement contenir des données d'une seule partie de message d'application. Une partie de message peut être fragmentée en un nombre quelconque de messages DUM avec le même paramètre AM-Part.

L'exemple suivant montre trois messages DUM contenant un message de réponse HTTP raccourci. La partie corps-de-réponse est fragmentée et envoyée dans deux messages DUM.

Exemple :

```
P: DUM 88 1 0
  Kept: 0
  AM-Part: en-tête-de-réponse

  64:HTTP/1.1 200 OK
  Content-Type: text/html
  Content-Length: 51

  ;
P: DUM 88 1 64
  Kept: 64
  AM-Part: corps-de-réponse

  19:<html><body>This is
  ;
P: DUM 88 1 83
  Kept: 83
  AM-Part: corps-de-réponse

  32: un simple message.</body></html>
  ;
```

3.5 Adaptation sélective

Le profil HTTP pour OCP s'applique à tous les messages HTTP. Cette portée inclut des messages HTTP comme des réponses 1xx (Information) des demandes POST, CONNECT, et OPTIONS, ainsi que des réponses avec des codes d'état d'extension et des demandes avec des méthodes d'extension. Sauf spécifiquement configuré pour faire autrement, un processeur OPES DOIT transmettre tous les messages HTTP pour adaptation aux serveurs d'invocation. Les instructions d'outrepassement d'OPES, les règles de traitement de message HTTP configurées, et la négociation OCP avec un serveur d'invocation sont tous des exemples d'une "configuration spécifique" acceptable qui font exception à cette règle.

Bien qu'il puisse paraître inutile de tenter d'adapter des messages de "contrôle" comme une réponse 100 (Continue) sauter de tels messages par défaut peut conduire à de sérieuses failles de la sécurité et à des problèmes d'interopérabilité. Par exemple, des informations industrielles sensibles peuvent être relayées via une réponse 100 Continue soigneusement étudiée ou une demande CONNECT malveillante peut n'être pas enregistrée si le processeur OPES ne transmet pas ces messages à un service d'invocation qui est supposé les traiter.

Par conception, la mise en œuvre de processeur OPES ne peut pas décider unilatéralement qu'un message HTTP ne vaut pas la peine de l'adaptation. Elle a besoin de l'opinion d'un serveur d'invocation, d'un réglage de configuration, ou d'une autre information externe pour prendre la décision.

3.6 En-têtes bond par bond

HTTP définit plusieurs en-têtes bond par bond (par exemple, Connection) et permet que des en-têtes d'extension soient spécifiés comme étant bond par bond (via le mécanisme d'en-tête Connection). Selon l'environnement et la configuration, un processeur OPES PEUT transmettre des en-têtes bond par bond aux serveurs d'invocation et PEUT utiliser des en-têtes bond par bond retournés par des serveurs d'invocation pour construire un message HTTP pour l'application de prochain bond. Cependant, voir au paragraphe 3.7 les exigences spécifiques de l'en-tête Transfer-Encoding.

Par exemple, un service de collecte d'enregistrements ou de statistiques peut vouloir voir les en-têtes bond par bond envoyés par le bond d'application précédent au processeur OPES et/ou les en-têtes bond par bond envoyés par le processeur OPES au prochain bond d'application. Un autre service peut en fait traiter la logique HTTP de suppression et ajout des en-

têtes bond par bond. De nombreux services vont ignorer les en-têtes bond par bond. La présente spécification ne définit pas de mécanisme pour distinguer ces cas d'utilisation.

3.7 Codages de transfert

Les messages HTTP peuvent utiliser des codages de transfert, qui est une caractéristique de codage bond par bond de HTTP. Les adaptations qui utilisent les codages de transfert HTTP doivent être explicitement négociées. La présente spécification ne documente pas de telles négociations. En l'absence d'une négociation explicite de codage de transfert, un agent OCP NE DOIT PAS envoyer de corps de message d'application avec un codage de transfert.

De façon informelle, la règle ci-dessus signifie que l'agent ou son environnement doivent s'assurer que tous les codages de transfert sont supprimés d'un corps de message HTTP avant qu'il entre dans la portée d'OCP. Un agent DOIT terminer la transaction OCP si il a à envoyer un corps de message d'application mais ne peut pas supprimer tous les codages de transfert. Les violations de ces règles conduisent à des problèmes d'interopérabilité.

Si un agent OCP reçoit des données d'application qui ont subi un codage de transfert en violation de l'exigence ci-dessus, il PEUT terminer la transaction OCP correspondante.

Un processeur OPES qui supprime les codages de transfert DOIT retirer l'en-tête Transfer-Encoding avant d'envoyer la partie en-tête au service d'invocation. Un serveur d'invocation qui reçoit un en-tête Transfer-Encoding PEUT supposer que les données d'application originales sont toujours en codage de transfert (et terminer la transaction). Le processeur OPES DOIT envoyer un en-tête Transfer-Encoding correct au prochain receveur HTTP, indépendamment de quel en-tête (si il en est) le serveur d'invocation a retourné.

L'enregistrement et l'écoute sont les exemples où la négociation des codages de transfert acceptables peut valoir la peine. Bien qu'un serveur d'invocation puisse n'être pas capable de supprimer un codage, il peut vouloir quand même enregistrer le message entier "tel quel". Dans la plupart des cas, cependant, le serveur d'invocation ne sera pas capable de traiter de façon significative des codages de transfert inconnus.

3.8 Correction de l'en-tête HTTP

Lorsque ils communiquent avec des applications HTTP, les processeurs OPES DOIVENT s'assurer de la correction de tous les en-têtes HTTP calculables documentés dans les spécifications auxquelles les processeurs entendent se conformer. Un en-tête calculable se définit comme un en-tête dont la valeur peut être calculée sur la base du seul corps de message. Par exemple, la correction des en-têtes Content-Length et Content-MD5 doit être vérifiée par les processeurs qui prétendent être conformes à HTTP/1.1 [RFC2616].

De façon informelle et par défaut, le processeur OPES doit valider et éventuellement recalculer, ajouter ou supprimer les en-têtes HTTP calculables afin de construire un message HTTP conforme à partir d'un message d'application adapté retourné par le serveur d'invocation. Si un certain processeur OPES fait confiance à certains en-têtes HTTP qu'un service d'invocation envoie, il peut utiliser ces en-têtes "tels quels".

Un processeur OPES PEUT transmettre un message HTTP partiellement adapté provenant d'un serveur d'invocation au prochain serveur d'invocation, sans vérifier que l'en-tête HTTP est correct. Par conséquent, un service d'invocation ne peut pas supposer que les en-têtes HTTP qu'il reçoit sont corrects ou finaux d'un point de vue HTTP.

Les paragraphes qui suivent présentent des lignes directrices pour le recalcul de certains en-têtes HTTP.

3.8.1 Nouveau calcul de taille de message

Par défaut, un agent OCP NE DOIT PAS faire confiance à l'en-tête Content-Length qui est envoyé au sein d'une partie d'en-tête de message HTTP. La longueur du message a pu être modifiée par un service d'invocation sans adaptation des en-têtes de message HTTP.

Avant d'envoyer le message HTTP à l'homologue HTTP, le processeur OPES doit s'assurer de la correction de l'indication de longueur du message conformément au paragraphe 4.4 de la [RFC2616].

À côté de s'assurer de la correction du message HTTP, les bons processeurs OPES établissent le message pour optimiser les performances, incluant de minimiser la latence de livraison. Précisément, indiquer la fin d'un message en fermant la connexion HTTP devrait être fait en dernier ressort :

- Si le serveur d'invocation envoie un paramètre AM-EL avec son message AMS, le processeur OPES DEVRAIT utiliser cette valeur pour créer un en-tête Content-Length pour être capable de garder une connexion HTTP persistante. Noter que les règles de HTTP interdisent d'utiliser un en-tête Content-Length dans les messages à codage de transfert.

- Si un paramètre AM-EL ou des informations équivalentes de longueur d'entité ne sont pas disponibles, et si les règles HTTP permettent un codage de transfert tronqué, le processeur OPES DEVRAIT utiliser le codage de transfert tronqué. Noter que tout en-tête Content-Length doit être retiré dans ce cas.

- Si la taille du message n'est pas connue a priori et si le codage de transfert tronqué ne peut pas être utilisé, mais si le processeur OPES peut attendre la fin de la transaction OCP avant de transmettre le message HTTP adapté sur une connexion HTTP persistante, le processeur DEVRAIT alors calculer et ajouter un en-tête Content-Length.

- Finalement, si toutes les optimisations sont non applicables, le processeur OPES DEVRAIT supprimer tout en-tête Content-Length et transmettre immédiatement les données adaptées, tout en indiquant la fin du message en fermant la connexion HTTP

3.8.2 En-tête Content-MD5

Par défaut, le processeur OPES DOIT supposer que le service d'invocation modifie le contenu d'une façon qui rend invalide la somme de contrôle MD5 du corps de message.

Conformément au paragraphe 14.15 de la [RFC2616], les intermédiaires HTTP ne doivent pas générer d'en-têtes Content-MD5. Un nouveau recalcul n'est donc possible que si le processeur OPES est considéré comme d'autorité pour l'entité qui est adaptée. Un processeur OPES qui n'est pas d'autorité DOIT retirer l'en-tête Content-MD5 sauf si il détecte que le message HTTP n'a pas été modifié ; dans ce cas, il PEUT laisser l'en-tête Content-MD5 dans le message. Quand une telle détection augmente significativement la latence du message, la suppression de l'en-tête Content-MD5 peut être une meilleure option.

3.9 Exemples

Voici un possible flux de messages OCP utilisant une demande de profil HTTP. Un usager final veut accéder à la page d'accueil de www.restricted.exemple.com, à travers le mandataire, mais l'accès est refusé par un URL qui bloque le service fonctionnant sur le serveur d'invocation utilisé par le mandataire.

Les messages OCP provenant du processeur OPES sont marqués "P:" et les messages OCP provenant du serveur d'invocation sont marqués "S:". La connexion OCP n'est pas fermée à la fin mais gardée ouverte pour la prochaine transaction OCP.

```
P: CS;
S: CS;
P: SGC 11 ({"31:ocp-test.exemple.com/url-filter"});
P: NO ({"53:http://www.iana.org/assignments/opes/ocp/http/request"})
  SG: 11
  ;
S: NR {"53:http://www.iana.org/assignments/opes/ocp/http/request"}
  SG: 11
  ;
P: TS 55 11;
P: AMS 55
  AM-EL: 0
  ;
P: DUM 55 0
  Kept: 0
  AM-Part: request-header
  235:GET http://www.restricted.exemple.com/ HTTP/1.1
  Accept: */*
  Accept-Language: de
  Accept-Encoding: gzip, deflate
  User-Agent: Mozilla/4.0 (compatible; Windows NT 5.0)
  Host: www.restricted.exemple.com
  Proxy-Connection: Keep-Alive
```

```
  ;
P: AME 55;
S: AMS 55;
S: DUM 55 0
  AM-Part: response-header
  76:HTTP/1.1 403 Interdit
  Content-Type: text/html
  Proxy-Connection: close

  ;
S: DUM 55 0
  AM-Part: response-body

  67:<html><body>L'accès à cette page
  vous est interdit.</body></html>
  ;
S: AME 55;
P: TE 55;
S: TE 55;
```

L'exemple suivant est une traduction de langage d'un petit fichier de texte en clair qui se trouve transféré dans une réponse HTTP. Dans cet exemple, les agents OCP négocient un profil pour toute la connexion OCP. La connexion OCP reste ouverte à la fin de la transaction OCP.

```
P: CS;
S: CS;
P: NO   ({"54:http://www.iana.org/assignments/opes/ocp/http/response"});
S: NR   {"54:http://www.iana.org/assignments/opes/ocp/http/response"};
P: SGC 12 ({"44:ocp-test.exemple.com/translate?from=EN&to=DE"});
P: TS 89 12;
P: AMS 89
  AM-EL: 86
  ;
P: DUM 89 0
  AM-Part: response-header

  65:HTTP/1.1 200 OK
  Content-Type: text/plain
  Content-Length: 86

  ;
P: DUM 89 65
  AM-Part: response-body

  86:Est il plus noble pour l'âme de souffrir
     Les coups et les flèches d'un destin outrageux
  ;
P: AME 89;
S: AMS 89
  AM-EL: 78
  ;
P: TE 89;
S: DUM 89 0
  AM-Part: response-header

  65:HTTP/1.1 200 OK
  Content-Type: text/plain
  Content-Length: 78

  ;
S: DUM 89 63
  AM-Part: response-body
```

```
   80:Ob's edler im Gemuet, die Pfeil und Schleudern
   des wuetenden Geschicks erdulden
   ;
S: AME 89;
S: TE 89;
```

L'exemple suivant montre la modification d'une ressource HTML et montre l'optimisation de la préservation des données. Le serveur d'invocation utilise un message DUY pour renvoyer une partie d'en-tête de réponse inchangée, mais comme il ne connaît pas la taille de la ressource HTML altérée au moment où il envoie le message AMS, le serveur d'invocation omet le paramètre AM-EL ; le processeur OPES est chargé d'ajuster l'en-tête Content-Length.

```
P: CS;
S: CS;
P: SGC 10 ({"30:ocp-test.exemple.com/ad-filter"});
P: NO ({"54:http://www.iana.org/assignments/opes/ocp/http/response"
   Aux-Parts: (request-header,request-body)
   },{"45:http://www.iana.org/assignments/opes/ocp/MIME"})
   SG: 10
   ;
S: NR {"54:http://www.iana.org/assignments/opes/ocp/http/response"
   Aux-Parts: (request-header)
   Content-Encodings: (gzip)
   }

   SG: 10
   ;
P: TS 88 10;
P: AMS 88
   AM-EL: 95
   ;
P: DUM 88 0
   AM-Part: request-header

   65:GET /opes/adsample.html HTTP/1.1
   Host: www.martin-stecher.de

   ;
P: DUM 88 65

   Kept: 65 64
   AM-Part: response-header

   64:HTTP/1.1 200 OK
   Content-Type: text/html
   Content-Length: 95

   ;
P: DUM 88 129
   Kept: 65 90
   AM-Part: response-body

   26:<html>
   <body>
   This is my
   ;
S: AMS 88;
P: DUM 88 155
   Kept: 65 158
   AM-Part: response-body

   68: new ad: <img src="my_ad.gif"
   width=88 height=31>
   </body>
```

```
  </html>
  ;
S: DUY 88 65 64
S: DPI 88 129 2147483647;
P: AME 88;
S: DUM 88 0
  AM-Part: response-body

  52:<html>
  <body>
  This is my new ad:
  </body>
  </html>
  ;
S: DPI 88 129 0;
P: TE 88;
S: AME 88;
S: TE 88;
```

4. Traçage

La [RFC3897] définit des facilités de traçage indifférentes à l'application dans OPES. La conformité à la présente spécification exige la conformité à la [RFC3897]. Lors de l'adaptation de HTTP, les entrées de trace sont fournies en utilisant les en-têtes de message HTTP. Les en-têtes d'extension HTTP suivantes sont définies pour porter les entrées de trace. Leur définition est donnée en utilisant la notation BNF et les éléments définis dans la [RFC2616].

```
OPES-System = "OPES-System" ":" #trace-entry
OPES-Via  = "OPES-Via" ":" #trace-entry
trace-entry = opes-agent-id *( ";" parameter )
opes-agent-id = absoluteURI
```

Un système OPES DOIT ajouter son entrée de trace à l'en-tête de système OPES. Les autres agents OPES DOIVENT utiliser l'en-tête OPES-Via si ils ajoutent leurs entrées de trace. Tous les agents OPES DOIVENT ajouter leurs entrées. De façon informelle, OPES-System est le seul en-tête de traçage OPES exigé tandis que OPES-Via fournit des détails facultatifs de traçage ; les deux en-têtes reflètent l'ordre des ajouts d'entrée de trace.

Si un en-tête OPES-Via est utilisé dans le message d'application original, un système OPES DOIT ajouter son entrée à l'en-tête OPES-Via. Autrement, un système OPES PEUT ajouter son entrée à l'en-tête OPES-Via. Si un système OPES utilise les deux en-têtes, il DOIT ajouter des entrées de trace identiques, sauf qu'il PEUT omettre certains paramètres d'entrée de trace ou tous de l'en-tête OPES-Via. De façon informelle, les entrées de système OPES dans l'en-tête OPES-Via sont utilisées pour délimiter et grouper les entrées OPES-Via provenant de différents systèmes OPES sans avoir une connaissance a priori des identifiants du système OPES.

Noter que tous ces en-têtes sont définis en utilisant la construction #list et donc, un message HTTP valide peut contenir plusieurs entrées de trace par en-tête. Les agents OPES DEVRAIENT utiliser un seul champ d'en-tête plutôt que d'utiliser plusieurs champs de même nom pour enregistrer une longue trace. Utiliser plusieurs champs d'en-tête d'extension de même nom est illégal du point de vue de HTTP et ne peut pas fonctionner avec certains mandataires HTTP sans capacité OPES.

Par exemple, voici un en-tête de message de réponse HTTP après que les adaptations OPES ont été appliquées par un seul processeur OPES exécutant 10 services OPES :

```
HTTP/1.1 200 OK
Date: Thu, 18 Sep 2003 06:25:24 GMT
Last-Modified: Wed, 17 Sep 2003 18:24:25 GMT
Content-type: application/octet-stream
OPES-System: http://www.cdn.exemple.com/opes?session=ac79a749f56
OPES-Via: http://www.cdn.exemple.com/opes?session=ac79a749f56,
   http://www.srvcs-4u.exemple.com/cat/?sid=123,
   http://www.srvcs-4u.exemple.com/cat/?sid=124,
   http://www.srvcs-4u.exemple.com/cat/?sid=125 ; mode=A
```

Dans l'exemple ci-dessus, le processeur OPES n'a pas inclus son entrée de trace ou son entrée de trace a été remplacée par une entrée de trace de système OPES. Seulement 3 sur les 10 services sont tracés. Les services restants n'incluaient pas leurs entrées ou leurs entrées ont été supprimées par le système ou processeur OPES. Le dernier service tracé incluait un paramètre "mode". Divers identifiants dans les entrées de trace n'auront probablement aucune signification pour le receveur du message, mais peuvent être décodés par le logiciel de système OPES.

Les entités OPES PEUVENT placer des entrées facultatives de traçage dans un en-queue de message (c'est-à-dire, des en-têtes d'entité à la fin d'un corps tronqué *(Chunked-Body)* d'un message en codage tronqué) pourvu que la présence de l'en-queue ne viole pas le protocole HTTP. Voir dans la [RFC3897] la définition de "entrée de traçage facultative". Les entités OPES NE DOIVENT PAS placer les entrées de traçage requises dans un en-queue de message.

5. Outrepassement

Un en-tête d'extension HTTP est introduit pour permettre un outrepassement de système OPES comme défini par la [RFC3897].

OPES-Bypass = "OPES-Bypass" ":" ("*" | 1#bypass-entry)
bypass-entry = opes-agent-id

Cet en-tête peut être ajouté aux demandes HTTP pour demander l'outrepassement du système OPES pour les agents OPES énumérés. Le caractère astérisque "*" est utilisé pour représenter tous les agents OPES possibles.

Voir dans la [RFC3897] ce qui peut être outrepassé et les exigences d'outrepassement.

6. Considérations relatives à l'IAB

Le traitement OPES des considérations du Bureau de l'Architecture de l'Internet (IAB) de l'IETF [RFC3238] est documenté dans "Traitement OPES des considérations de l'IAB" [RFC3914].

7. Considérations sur la sécurité

Les considérations sur la sécurité indépendantes de l'application sont documentées dans les spécifications OPES indifférentes aux applications. Les profils HTTP n'introduisent aucune considération sur la sécurité spécifique de HTTP. Cependant, cela n'implique pas que les adaptations HTTP soient à l'épreuve des menaces sur la sécurité.

Des exemples de menaces spécifiques incluent des adaptations comme la réécriture de l'URI de demande d'une demande CONNECT HTTP ou la suppression d'un en-tête Upgrade bond par bond HTTP avant que le mandataire HTTP puisse agir sur lui. Comme avec toute adaptation, l'agent OPES NE DOIT PAS effectuer de telles actions sans le consentement du client ou serveur HTTP.

8. Considérations relatives à l'IANA

L'IANA enregistre les caractéristiques de profil de demande et de réponse (paragraphe 3.2) en utilisant la procédure d'enregistrement mentionnée dans la Section "Considérations relatives à l'IANA" du cœur OCP [RFC4037]. Les paramètres "uri" correspondants pour les deux caractéristiques sont :

- http://www.iana.org/assignments/opes/ocp/http/request
- http://www.iana.org/assignments/opes/ocp/http/response

9. Conformité

La conformité aux mécanismes OPES est définie dans les spécifications indifférentes aux applications correspondantes. Les profils HTTP pour ces mécanismes utilisent les définitions de conformité correspondantes à partir de ces spécifications, comme si chaque profil était incorporé dans la spécification indifférente à l'application qu'elle profile.

10. Références

10.1 Références normatives

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. *(MàJ par* RFC8174)

[RFC2616] R. Fielding et autres, "Protocole de transfert hypertexte -- HTTP/1.1", juin 1999. *(D.S., MàJ par* 2817, 6585)

[RFC3897] A. Barbir, "Communication des entités et des points d'extrémité des services marginaux à connexion libre (OPES)", septembre 2004. *(Information)*

[RFC4037] A. Rousskov, "Cœur du protocole d'invocation (OCP) des services marginaux à connexion libre (OPES)", mars 2005. *(P.S.)*

10.2 Références pour information

[RFC3238] S. Floyd, L. Daigle, "Considérations architecturales et de politique de l'IAB pour des services marginaux à connexion libre (OPES)", janvier 2002. *(Information)*

[RFC3752] A. Barbir et autres, "Services marginaux à connexion libre (OPES) : cas d'utilisation et scénarios de développement", avril 2004. *(Information)*

[RFC3835] A. Barbir et autres, "Architecture pour les services marginaux à connexion libre (OPES)", août 2004. *(Information)*

[RFC3836] A. Beck et autres, "Exigences pour les protocoles d'invocation de services marginaux à connexion libre (OPES)", août 2004. *(Information)*

[RFC3837] A. Barbir et autres, "Menaces et risques pour la sécurité des services marginaux à connexion libre (OPES)", août 2004. *(Information)*

[RFC3838] A. Barbir et autres, "Exigences de politique, d'autorisation, et de mise en application des services marginaux à connexion libre (OPES)", août 2004. *(Information)*

[RFC3914] A. Barbir, A. Rousskov, "Traitement des considérations de l'IAB par les services marginaux à connexion libre (OPES)", octobre 2004. *(Information)*

[rules-p] Beck, A. et A. Rousskov, "P: Message Processing Language", projet non suivi, octobre 2003.

8. Remerciements

Les auteurs remercient chaleureusement de leurs contributions Robert Collins (Syncretize) et Larry Masinter (Adobe). Larry Masinter a fait une relecture précoce du présent document.

Adresse des auteurs

Alex Rousskov
The Measurement Factory
mél : rousskov@measurement-factory.com
URI : http://www.measurement-factory.com/

Martin Stecher
CyberGuard Corporation
Vattmannstr. 3
Paderborn 33100
DE
mél : martin.stecher@webwasher.com
URI : http://www.webwasher.com/

Déclaration complète de droits de reproduction

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