Groupe de travail Réseau
Request for Comments : 3836
Catégorie : Information

Traduction Claude Brière de L'Isle

A. Beck & M. Hofmann, Lucent Technologies
H. Orman, Purple Streak Development
R. Penno, Nortel Networks
A. Terzis, Johns Hopkins University
août 2004

Exigences pour les protocoles d'invocation des services marginaux à connexion libre (OPES)

Statut de ce mémoire
Le présent document apporte des informations pour la communauté de l'Internet. Il ne spécifie aucune sorte de norme de l'Internet. La distribution du présent mémoire n'est soumise à aucune restriction.

Résumé
Le présent document spécifie les exigences que le protocole d'invocation des services marginaux à connexion libre (OPES, *Open Pluggable Edge Services*) doit satisfaire afin de prendre en charge l'exécution à distance des services OPES. Les exigences sont destinées à aider à évaluer les protocoles candidats possibles, ainsi qu'à guider le développement de tels protocoles.

Table des matières

1. Terminologie

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. Introduction

L'architecture des services marginaux à connexion libre (OPES, *Open Pluggable Edge Services*) [RFC3835] permet des services d'application coopératifs (services OPES) entre un fournisseur de données, un consommateur de données, et zéro, un ou plusieurs processeurs OPES. Les services d'application considérés analysent, et éventuellement transforment, des messages de niveau application échangés entre le fournisseur de données et le consommateur de données.

L'exécution de tels services est gouvernée par un ensemble de règles installées sur le processeur OPES. L'application des règles peut déclencher l'exécution d'applications de services locales sur le processeur OPES. Autrement, le processeur OPES peut distribuer la responsabilité de l'exécution du service en communiquant et en collaborant avec un ou plusieurs serveurs d'invocation distants. Comme décrit dans la [RFC3835], un processeur OPES communique avec, et invoque des services sur un serveur d'invocation en utilisant un protocole d'invocation. Le présent document présente les exigences pour un tel protocole.

Les exigences dans ce document sont divisées en trois catégories - exigences fonctionnelles, exigences de performances, et exigences de sécurité. Chaque exigence est présentée comme une ou plusieurs déclarations, suivie par un bref matériel explicatif comme approprié.

3. Exigences fonctionnelles

3.1 Fiabilité

Le protocole d'invocation OPES DOIT être capable de fournir une fiabilité ordonnée pour la communication entre un processeur OPES et un serveur d'invocation. De plus, le protocole d'invocation DEVRAIT être capable de fournir une fiabilité non ordonnée.

Afin de satisfaire les exigences de fiabilité, le protocole d'invocation DEVRAIT spécifier qu'il doit être utilisé avec un protocole de transport qui fournit une fiabilité ordonnée/non ordonnée à la couche transport, par exemple TCP [RFC0793] ou SCTP [RFC2960].

3.2 Évitement d'encombrement

Le protocole d'invocation OPES DOIT assurer qu'un évitement d'encombrement correspondant au standard de la [RFC2914] est appliqué sur toutes les communications entre le processeur OPES et le serveur d'invocation. À cette fin, le protocole d'invocation DEVRAIT utiliser un protocole de contrôle d'encombrement de couche transport, soit TCP [RFC0793], soit SCTP [RFC2960].

3.3 Transactions d'invocation

Le protocole d'invocation OPES DOIT permettre à un processeur OPES et un serveur d'invocation d'effectuer des transactions d'invocation dans le but d'échanger des messages de protocole de niveau application partiel ou complet (ou de leurs modifications). Plus précisément, le protocole d'invocation DOIT permettre à un processeur OPES de transmettre un message d'application partiel ou complet à un serveur d'invocation de sorte qu'un ou plusieurs services OPES puissent traiter le message d'application transmis (ou des parties de celui-ci). Le résultat de l'opération de service peut être un message d'application modifié. Le protocole d'invocation DOIT donc permettre au serveur d'invocation de retourner un message d'application modifié ou les parties modifiées d'un message d'application au processeur OPES. De plus, le protocole d'invocation DOIT permettre à un serveur d'invocation de faire rapport du résultat d'une transaction d'invocation (par exemple, sous la forme d'un code d'état) au processeur OPES.

Une transaction d'invocation est définie comme un échange de messages entre un processeur OPES et un serveur d'invocation consistant en une demande d'invocation et une réponse d'invocation. La demande d'invocation et la réponse d'invocation PEUVENT toutes deux consister chacune en un ou plusieurs messages de protocole d'invocation, c'est-à-dire

une série de messages de protocole. Une demande d'invocation DOIT toujours contenir un message d'application partiel ou complet. Une réponse d'invocation DOIT toujours indiquer le résultat de la transaction d'invocation. Une réponse d'invocation PEUT contenir un message d'application modifié.

Les transactions d'invocation sont toujours initiées par une demande d'invocation provenant d'un processeur OPES et sont normalement terminées par une réponse d'invocation provenant d'un serveur d'invocation. Le protocole d'invocation OPES DOIT, cependant, fournir aussi un mécanisme qui permette à l'un ou l'autre point d'extrémité d'une transaction d'invocation de terminer une transaction d'invocation avant qu'une demande ou réponse d'invocation ait été complètement reçue par le point d'extrémité d'invocation correspondant. Un tel mécanisme DOIT assurer qu'une terminaison prématurée d'une transaction d'invocation ne résulte pas en la perte des données du message d'application.

Une terminaison prématurée d'une transaction d'invocation est requise pour prendre en charge les services OPES, qui peuvent se terminer même avant qu'ils aient traité le message d'application entier. Les services d'analyse de contenu, par exemple, peuvent être capables de classer un objet de la Toile après avoir traité juste les tout premiers octets de l'objet.

3.4 Connexions d'invocation

Le protocole d'invocation OPES DOIT permettre à un processeur OPES et à un serveur d'invocation d'effectuer plusieurs transactions d'invocation sur une connexion d'invocation. De plus, le protocole d'invocation DOIT fournir une méthode pour associer les transactions d'invocation aux connexions d'invocation. Une connexion d'invocation est définie comme une connexion logique à la couche application entre un processeur OPES et un serveur d'invocation. Une connexion d'invocation PEUT avoir certains paramètres associés, par exemple, des paramètres qui contrôlent le comportement de reprise sur défaillance des points d'extrémité de connexion. Des paramètres spécifiques de la connexion PEUVENT être négociés entre processeurs OPES et serveurs d'invocation (voir au paragraphe 3.12).

Le protocole d'invocation OPES PEUT choisir de multiplexer plusieurs connexions d'invocation sur une seule connexion de couche transport si un mécanisme de contrôle de flux garantissant un traitement équitable parmi les connexions d'invocation multiplexées est appliqué.

Les connexions d'invocation DOIVENT toujours être initiées par un processeur OPES. Une connexion d'invocation PEUT être fermée par l'un ou l'autre point d'extrémité de la connexion, pourvu que cela n'affecte pas le fonctionnement normal des transactions d'invocation en cours associées à la connexion d'invocation.

3.5 Échange de message asynchrone

Le protocole d'invocation OPES DOIT prendre en charge un échange de messages asynchrone sur les connexions d'invocation.

Afin de permettre un traitement asynchrone sur le processeur OPES et le serveur d'invocation, il DOIT être possible de séparer la production de demande du traitement de réponse. Le protocole DOIT donc permettre plusieurs demandes d'invocation en même temps et fournir une méthode pour corréler les réponses d'invocation avec les demandes.

De plus, le protocole d'invocation DOIT permettre à un serveur d'invocation de répondre à une demande d'invocation avant qu'il ait reçu la demande entière.

3.6 Segmentation de message

Le protocole d'invocation OPES DOIT permettre à un processeur OPES de transmettre un message d'application à un serveur d'invocation dans une série de plus petits fragments de message. Le protocole d'invocation DOIT de plus permettre au serveur d'invocation receveur de ré-assembler les fragments du message d'application.

De même, le protocole d'invocation DOIT permettre à un serveur d'invocation de retourner un message d'application à un processeur OPES dans une série de plus petits fragments de message. Le protocole d'invocation DOIT permettre au processeur OPES receveur de ré-assembler les fragments du message d'application.

Selon le protocole de couche application utilisé sur le chemin des données, les messages d'application peuvent être de très grande taille (par exemple dans le cas de flux audio/vidéo) ou d'une taille inconnue. Dans les deux cas, le processeur OPES doit initier une transaction d'invocation avant qu'il ait reçu le message d'application entier pour éviter de longs délais pour le consommateur de données. Le processeur OPES DOIT donc être capable de transmettre des fragments ou tronçons d'un message d'application à un serveur d'invocation lorsque il les reçoit du fournisseur ou consommateur de données. De

même, le serveur d'invocation DOIT être capable de traiter et retourner des fragments de message d'application lorsque il les reçoit du processeur OPES.

La segmentation du message d'application est aussi requise si le protocole d'invocation OPES fournit un mécanisme de contrôle de flux afin de multiplexer plusieurs connexions d'invocation sur une seule connexion de couche transport (voir au paragraphe 3.4).

3.7 Prise en charge du mécanisme de garde en vie

Le protocole d'invocation OPES DOIT fournir un mécanisme de garde en vie qui, si il est utilisé, va permettre aux deux points d'extrémité d'une connexion d'invocation de détecter une défaillance de l'autre point d'extrémité, même en l'absence de transactions d'invocation. Le protocole d'invocation PEUT spécifier que les messages de garde en vie soient échangés sur les connexions d'invocation existantes ou sur une connexion séparée entre le processeur OPES et le serveur d'invocation. Le protocole d'invocation PEUT aussi spécifier que l'utilisation du mécanisme de garde en vie est facultatif.

La détection de la défaillance d'un serveur d'invocation peut permettre à un processeur OPES d'établir une connexion d'invocation avec un serveur d'invocation en attente afin que les futures transactions d'invocation ne résultent pas en la perte des données du message d'application. La détection de la défaillance d'un processeur OPES peut permettre à un serveur d'invocation de libérer les ressources qui autrement n'auraient pas été disponibles pour des transactions d'invocation avec d'autres processeurs OPES.

3.8 Fonctionnement dans un environnement de NAT

Le protocole OPES DEVRAIT être neutre à l'égard des traducteurs d'adresse réseau (NAT), c'est-à-dire, que son fonctionnement ne devrait pas être compromis par la présence d'un ou plusieurs appareils de NAT sur le chemin entre un processeur OPES et un serveur d'invocation.

3.9 Plusieurs serveurs d'invocation

Le protocole d'invocation OPES DOIT permettre à un processeur OPES de communiquer simultanément avec plus d'un serveur d'invocation.

Dans les plus grands réseaux, les services OPES seront probablement hébergés par différents serveurs d'invocation. Donc, un processeur OPES aura probablement à communiquer avec plusieurs serveurs d'invocation. La conception du protocole DOIT permettre à un processeur OPES de le faire.

3.10 Plusieurs processeurs OPES

Le protocole d'invocation OPES DOIT permettre à un serveur d'invocation de communiquer simultanément avec plus d'un processeur OPES.

La conception du protocole DOIT prendre en charge un scénario dans lequel plusieurs processeurs OPES utilisent les services d'un seul serveur d'invocation.

3.11 Prise en charge de différents protocoles d'application

Le protocole d'invocation OPES DEVRAIT être neutre à l'égard du protocole d'application, c'est-à-dire, il NE DEVRAIT PAS faire d'hypothèses sur les caractéristiques du protocole de couche application utilisé sur le chemin des données entre le fournisseur de données et le consommateur de données. Au minimum, le protocole d'invocation DOIT être compatible avec HTTP [RFC2816].

Les entités OPES sur le chemin des données peuvent utiliser des protocoles de couche application différents, incluant, sans s'y limiter, HTTP [RFC2816] et RTP [RFC3550]. Il serait souhaitable d'être capable d'utiliser le même protocole d'invocation OPES pour chacun de ces protocoles de couche application.

3.12 Négociations de capacités et paramètres

Le protocole d'invocation OPES DOIT prendre en charge la négociation de capacités et paramètres de connexion d'invocation entre un processeur OPES et un serveur d'invocation. Cela implique que le processeur OPES et le serveur

d'invocation DOIVENT être capables d'échanger leurs capacités et préférences. Ils DOIVENT ensuite être capables de s'engager dans un procès de négociation déterministe qui se termine soit par un accord des deux points d'extrémité sur les capacités et paramètres à utiliser pour les futures connexions/transactions d'invocation, soit par la détermination que leurs capacités sont incompatibles.

Les capacités et paramètres qui pourraient être négociés entre un processeur OPES et un serveur d'invocation incluent (sans s'y limiter) : la version de protocole d'invocation, le comportement de reprise sur défaillance, le débit de battement de cœur pour les messages de garde en vie, les paramètres relatifs à la sécurité, etc.

Le protocole d'invocation NE DOIT PAS utiliser la négociation pour déterminer le protocole de transport à utiliser pour les connexions d'invocation. Le protocole d'invocation PEUT, cependant, spécifier qu'un certain protocole de message d'application (par exemple, HTTP [RFC2816], RTP [RFC3550]) exige l'utilisation d'un certain protocole de transport (par exemple, TCP [RFC0793], SCTP [RFC2960]).

Les paramètres de connexion d'invocation peuvent aussi relever des caractéristiques des services d'invocation OPES si par exemple, les connexions d'invocation sont associées à un ou plusieurs services OPES spécifiques. Un paramètre spécifique de service OPES peut, par exemple, spécifier quelles parties d'un message d'application sont nécessaires pour le fonctionnement d'un service OPES.

Les paramètres de connexion d'invocation DOIVENT être négociés sur la base de la connexion d'invocation et avant qu'aucune transaction d'invocation soit effectuée sur la connexion d'invocation correspondante. D'autres paramètres et capacités, comme le comportement de reprise sur défaillance, PEUVENT être négociés entre les deux points d'extrémité indépendamment des connexions d'invocation.

Les parties à un protocole d'invocation PEUVENT utiliser des connexions d'invocation pour négocier tout ou partie de leurs capacités et paramètres. Autrement, une connexion de contrôle séparée PEUT être utilisée à cette fin.

3.13 Métadonnées et instructions

Le protocole d'invocation OPES DOIT fournir un mécanisme pour que les points d'extrémité d'une transaction d'invocation particulière incluent des métadonnées et instructions pour le processeur OPES ou le serveur d'invocation dans les demandes et réponses d'invocation.

Précisément, le protocole d'invocation DOIT permettre à un processeur OPES d'inclure des informations sur le message d'application transmis dans une demande d'invocation, par exemple afin de spécifier le type de messages d'application transmis ou de spécifier quelles parties du message d'application sont transmises au serveur d'invocation. De même, le serveur d'invocation DOIT être capable d'inclure des informations sur le message d'application retourné.

Le processeur OPES DOIT de plus être capable d'inclure une liste ordonnée d'un ou plusieurs services OPES spécifiés de façon univoque qui sont à effectuer sur le message d'application transmis dans l'ordre spécifié. Cependant,comme le protocole d'invocation PEUT aussi choisir d'associer les connexions d'invocation à des services OPES spécifiques, il peut n'être pas besoin d'identifier les services OPES sur la base de la transaction d'invocation.

De plus, le protocole d'invocation OPES DOIT permettre au serveur d'invocation d'indiquer au processeur OPES l'accessibilité des réponses d'invocation. Cela implique que les réponses d'invocation puissent avoir à porter des instructions de contrôle de mise en antémémoire pour le processeur OPES.

Le protocole d'invocation OPES DOIT de plus permettre au processeur OPES d'indiquer au serveur d'invocation si il a conservé une copie locale du message d'application transmis (ou de certaines de ses parties). Ces informations permettent au serveur d'invocation de déterminer si le message d'application transmis doit être retourné au processeur OPES, même si il n'a pas été modifié par un service OPES.

Le protocole d'invocation OPES DOIT aussi permettre aux processeurs OPES de se conformer aux exigences de traçage de l'architecture OPES telles qu'exposées dans les [RFC3835] et [RFC3238]. Cela implique que le protocole d'invocation DOIT permettre à un serveur d'invocation de porter au processeur OPES des informations sur les opérations de service OPES effectuées sur le message d'application transmis.

4. Exigences de performances

4.1 Efficacité du protocole

Le protocole d'invocation OPES DEVRAIT avoir une latence minimale. Par exemple, la taille et la complexité de ses en-têtes pourrait être réduite au minimum.

Parce que les transactions d'invocation OPES ajoutent de la latence aux transactions de protocole d'application sur le chemin des données, l'efficacité du protocole d'invocation est cruciale pour les performances globales.

5. Exigences de sécurité

En l'absence de tout mécanisme de sécurité, des informations sensibles pourraient être communiquées entre le processeur OPES et le serveur d'invocation en violation de la politique de sécurité et de confidentialité de l'un et l'autre point d'extrémité, par mauvaise configuration ou par une attaque délibérée d'un infiltré. En utilisant une forte authentification, le chiffrement du message, et des vérifications d'intégrité, cette menace peut être minimisée à un petit ensemble d'infiltrés et/ou d'erreur de configuration de l'opérateur.

Le processeur OPES et les serveurs d'invocation DEVRAIENT avoir des politiques applicables qui limitent les parties avec lesquelles ils communiquent et qui déterminent les protections à utiliser sur la base des identités des points d'extrémité et autres données (telles que les politiques d'utilisateur final). Afin d'appliquer ces politiques, il DOIT être capable d'authentifier les points d'extrémité du protocole d'invocation en utilisant des méthodes de chiffrement.

5.1 Authentification, confidentialité, et intégrité

Les parties au protocole d'invocation DOIVENT avoir de bonnes bases pour lier les identités authentifiées aux points d'extrémité de protocole, et elles DOIVENT vérifier que ces identités sont cohérentes avec leur politique de sécurité.

Le protocole d'invocation OPES DOIT assurer l'authentification, la confidentialité, et l'intégrité du message entre le processeur OPES et le serveur d'invocation. Il DOIT assurer l'authentification mutuelle. À cette fin, le protocole d'invocation DEVRAIT utiliser les mécanismes de sécurité existants. La spécification du protocole d'invocation n'est pas obligée de spécifier les mécanismes de sécurité, mais PEUT à la place se référer à un protocole de sécurité de niveau inférieur et expliquer comment ses mécanismes sont à utiliser avec le protocole d'invocation.

5.2 Confidentialité bond par bond

Si le chiffrement bond par bond est une exigence pour le chemin du contenu, cette confidentialité DOIT alors être étendue à la communication entre le processeur OPES et le serveur d'invocation. Bien qu'il soit recommandé que la communication entre le processeur OPES et le serveur d'invocation soit toujours chiffrée, le chiffrement PEUT être facultatif si le processeur OPES et le serveur d'invocation sont tous deux co-localisés sur un seul domaine administratif avec de fortes garanties de confidentialité.

Afin de minimiser l'exposition des données, le protocole d'invocation DOIT utiliser une clé de chiffrement différente pour chaque flux de contenu chiffré.

5.3 Fonctionnement dans des domaines qui ne sont pas de confiance

Le protocole d'invocation OPES DOIT opérer en toute sécurité à travers des domaines qui ne sont pas de confiance entre le processeur OPES et le serveur d'invocation.

Si les canaux de communication entre le processeur OPES et le serveur d'invocation s'étendent en dehors de l'organisation qui est responsable des services OPES, l'authentification et la protection de message (confidentialité et intégrité) du point d'extrémité DOIVENT être utilisées.

5.4 Confidentialité

Toute communication portant des informations relevant des politiques de confidentialité DOIT protéger les données en utilisant le chiffrement.

6. Considérations sur la sécurité

Les exigences de sécurité pour le protocole d'invocation OPES sont discutées à la Section 5.

7. Références

7.1 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)

[RFC2818] E. Rescorla, "HTTP sur TLS", mai 2000. *(Information)*

[RFC2914] S. Floyd, "Principes du contrôle d'encombrement", BCP 41, septembre 2000.

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

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

7.2 Références pour information

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

[RFC2960] R. Stewart et autres, "Protocole de transmission de commandes de flux", octobre 2000. (*Obsolète, voir* RFC4960) *(P.S.)*

[RFC3550] H. Schulzrinne, S. Casner, R. Frederick et V. Jacobson, "RTP : un protocole de transport pour les applications en temps réel", STD 64, juillet 2003. *(MàJ par* RFC7164, RFC7160, RFC8083, RFC8108)

8. Remerciements

Des parties du présent document se fondent sur un travail antérieur de Anca Dracinschi Sailer, Volker Hilt, et Rama R. Menon. Les auteurs tiennent à remercier les participants au groupe de travail OPES de leurs commentaires sur le document.

9. Adresse des auteurs

Andre Beck
Lucent Technologies
101 Crawfords Corner Road
Holmdel, NJ 07733
US
mél : abeck@bell-labs.com

Markus Hofmann
Lucent Technologies
Room 4F-513
101 Crawfords Corner Road
Holmdel, NJ 07733
téléphone : +1 732 332 5983
mél : hofmann@bell-labs.com

Hilarie Orman
Purple Streak Development
mél : ho@alum.mit.edu
URI : http://www.purplestreak.com

Reinaldo Penno
Nortel Networks
00 Technology Park Drive
Billerica, MA 01821
US
mél : rpenno@nortelnetworks.com

Andreas Terzis
Computer Science Department
Johns Hopkins University
3400 North Charles Street, 224 NEB
Baltimore, MD 21218
téléphone : +1 410 516 5847
mél : terzis@cs.jhu.edu

10. Déclaration complète de droits de reproduction

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