

Groupe de travail Réseau
Request for Comments : 3841
 Catégorie : En cours de normalisation
 Traduction Claude Brière de L'Isle

J. Rosenberg, dynamicsoft
 H. Schulzrinne, Columbia University
 P. Kyzivat, Cisco Systems
 août 2004

Préférences de l'appelant pour le protocole d'initialisation de session (SIP)

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 (2004).

Résumé

Le présent document décrit un ensemble d'extensions au protocole d'initialisation de session (SIP, *Session Initiation Protocol*) qui permet à un appelant d'exprimer ses préférences au sujet du traitement des demandes dans les serveurs. Ces préférences incluent la capacité de choisir à quels identifiants de ressource universelle (URI, *Uniform Resource Identifier*) est acheminée une demande, et de spécifier certaines directives de traitement de demande aux mandataires et serveurs de redirection. Cela est fait en définissant trois nouveaux champs d'en-tête de demande, Accept-Contact, Reject-Contact, et Request-Disposition, qui spécifient les préférences de l'appelant.

Table des Matières

1. Introduction.....	1
2. Terminologie.....	2
3. Définitions.....	2
4. Vue d'ensemble du fonctionnement.....	3
5. Comportement de l'UAC.....	3
5.1 Préférences de traitement de la demande.....	4
5.2 Préférences d'ensemble de caractéristiques.....	4
6. Comportement de l'UAS.....	5
7. Comportement du mandataire.....	5
7.1 Traitement de Request-Disposition.....	5
7.2 Correspondance de préférence et de capacité.....	5
8. Transposition des paramètres de caractéristique en prédicat.....	10
9. Définitions de champ d'en-tête.....	11
9.1. Request-Disposition.....	12
9.2 Champs d'en-tête Accept-Contact et Reject-Contact.....	12
10. BNF augmenté.....	12
11. Considérations pour la sécurité.....	13
12. Considérations relatives à l'IANA.....	13
13. Remerciements.....	13
14. Références.....	13
14.1 Références normatives.....	13
14.2 Références pour information.....	14
15. Adresse des auteurs.....	14
16. Déclaration complète de droits de reproduction.....	14

1. Introduction

Lorsque un serveur du protocole d'initialisation de session (SIP) [RFC3261] reçoit une demande, il peut prendre un certain nombre de décisions en ce qui concerne le traitement de la demande. Cela inclut :

- o de passer par un mandataire ou de rediriger la demande,

- o sur quels URI donner mandat ou rediriger,
- o de subdiviser ou non,
- o de faire ou non une recherche récurrente,
- o de faire une recherche en parallèle ou en séquence.

Le serveur peut fonder ces décisions sur toute politique locale. Cette politique peut être configurée de façon statique, ou peut se fonder sur l'exécution d'un programme ou d'un accès à une base de données.

Cependant, l'administrateur du serveur n'est pas la seule entité qui a intérêt au traitement de la demande. Il y a au moins trois parties qui y ont intérêt : (1) l'administrateur du serveur, (2) l'utilisateur qui envoie la demande, et (3) l'utilisateur à qui la demande est adressée. Les directives de l'administrateur sont incorporées dans la politique du serveur. Les préférences de l'utilisateur à qui la demande est adressée (auquel on se réfère comme étant l'appelé, même si la méthode de demande peut n'être pas INVITE) peuvent être très facilement exprimées par un scénario écrit dans un certain type de langage écrit, comme le langage de traitement d'appel (CPL, *Call Processing Language*) [RFC2824]. Cependant, il n'existe aucun mécanisme pour incorporer les préférences de l'utilisateur qui envoie la demande (aussi désigné comme l'appelant, même si la méthode de la demande peut n'être pas INVITE). Par exemple, l'appelant pourrait vouloir parler à un utilisateur spécifique, mais ne veut le joindre qu'au travail, parce que l'appel est un appel professionnel. Autre exemple, l'appelant pourrait vouloir joindre un utilisateur, mais pas sa messagerie vocale, car il est important que l'appelant parle à l'appelé. Dans ces deux exemples, la préférence de l'appelant revient à ce qu'un mandataire fasse un choix d'acheminement particulier sur la base des préférences de l'appelant.

La présente extension permet à l'appelant de voir satisfaites ces préférences. Elle le fait en spécifiant des mécanismes par lesquels un appelant peut fournir des préférences sur le traitement d'une demande. Il y a deux types de préférences. L'un d'eux, appelé préférences de traitement de demande, est encapsulé dans le champ d'en-tête Request-Disposition. Il donne des directives spécifiques du traitement des demandes pour un serveur. L'autre, appelé préférences de caractéristiques, est présent dans les champs d'en-tête Accept-Contact et Reject-Contact. Il permet à l'appelant de fournir un ensemble de caractéristiques [RFC2533] qui exprime ses préférences quant aux caractéristiques de l'UA à atteindre. Celles-ci sont confrontées à un ensemble de caractéristiques fourni par l'UA à son registraire [RFC3840]. L'extension est d'objet très général, et n'est pas liée à un service particulier. C'est plutôt un outil qui peut être utilisé dans le développement de nombreux services.

Un exemple de service permis par les préférences de l'appelant est un service de "numéro unique". Un utilisateur peut avoir une seule identité (son URI SI) pour tous ses appareils – son téléphone cellulaire, son PDA, son téléphone professionnel, son téléphone personnel, et ainsi de suite. Si l'appelant veut joindre l'utilisateur sur son téléphone professionnel, il choisit simplement "téléphone professionnel" sur un menu déroulant d'options lorsque il appelle cet URI. Les utilisateurs n'auront plus besoin de tenir et distribuer des identités différentes pour chaque appareil.

2. Terminologie

Dans le présent document, les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDÉ", "PEUT", et "FACULTATIF" sont à interpréter comme décrit dans le BCP 14, [RFC 2119] et indiquent les niveaux d'exigence pour les mises en œuvre conformes.

3. Définitions

Beaucoup de la terminologie utilisée dans la présente spécification est présentée dans la [RFC3840]. La présente spécification définit les termes supplémentaires suivants :

Appelant : dans le contexte de la présente spécification, appelant se réfère à l'utilisateur au nom duquel fonctionne un UAC. Il ne se limite pas à l'utilisateur dont l'UAC envoie une demande INVITE.

Préférences de caractéristiques : ce sont les préférences de l'appelant qui décrivent les propriétés désirées d'un UA auquel la demande doit être acheminée. Les préférences de caractéristiques peuvent être rendues explicites avec les champs d'en-tête Accept-Contact et Reject-Contact.

Préférences de traitement de demande : préférences de l'appelant qui décrivent le traitement désiré de la demande à un serveur. Ces préférences sont portées dans le champ d'en-tête Request-Disposition.

Ensemble cible : c'est un ensemble de candidats URI auxquels un serveur mandataire ou de redirection peut envoyer ou rediriger une demande. Les ensembles cibles sont fréquemment obtenus d'un enregistrement, mais pas

nécessairement.

Préférence explicite : préférence de l'appelant indiquée explicitement dans les champs d'en-tête Accept-Contact ou Reject-Contact.

Préférence implicite : préférence de l'appelant qui est impliquée par la présence d'autres aspects d'une demande. Par exemple, si la méthode de la demande est INVITE, elle représente une préférence implicite de l'appelant pour acheminer la demande à un UA qui prend en charge la méthode INVITE.

4. Vue d'ensemble du fonctionnement

Lorsque un appelant envoie une demande, il peut facultativement inclure de nouveaux champs d'en-tête qui demandent un certain traitement au serveur. Ces préférences entrent dans deux catégories. La première catégorie, appelée préférences de traitement de la demande, est portée dans le champ d'en-tête Request-Disposition. Elle décrit un comportement spécifique qui est désiré au serveur. Les préférences de traitement de la demande incluent de savoir si l'appelant souhaite que le serveur ait un mandataire ou redirige, et si une recherche séquentielle ou en parallèle est désirée. Ces préférences peuvent être appliquées à chaque serveur mandataire ou de redirection sur le chemin de signalisation de l'appel.

La seconde catégorie de préférences, appelée préférences de caractéristiques, est portée dans les champs d'en-tête Accept-Contact et Reject-Contact. Ces champs d'en-tête contiennent des ensembles de caractéristiques, représentés par les mêmes paramètres de caractéristiques que ceux utilisés pour indiquer les capacités [RFC3840]. Ici, les paramètres de caractéristiques représentent les préférences de l'appelant. Le champ d'en-tête Accept-Contact contient des ensembles de caractéristiques qui décrivent les UA que l'appelant aimerait joindre. Le champ d'en-tête Reject-Contact contient des ensembles de caractéristiques qui, si elles correspondent à celui d'un UA, impliquent que la demande ne devrait pas être acheminée à cet UA.

Les mandataires utilisent les informations des champs d'en-tête Accept-Contact et Reject-Contact pour faire un choix parmi les contacts de leur ensemble cible. Lorsque aucun des ces champs d'en-tête n'est présent, le mandataire calcule les préférences implicites à partir de la demande. Ce sont des préférences d'appelant qui ne sont pas placées explicitement dans la demande, mais peuvent être déduites de la présence d'autres composants du message. Par exemple, si la méthode de la demande est INVITE, c'est une préférence implicite d'acheminer l'appel sur un UA qui prend en charge la méthode INVITE.

Les deux préférences de traitement de la demande et de caractéristiques peuvent apparaître dans toute demande, pas seulement INVITE. Cependant, elles ne sont utiles dans les demandes que lorsque les mandataires ont besoin de déterminer une cible de demande. Si le domaine dans l'URI de demande n'est possédé par aucun mandataire le long du chemin de la demande, ces mandataires ne vont jamais accéder à un service de localisation, et donc, n'auront jamais l'opportunité d'appliquer les préférences de l'appelant. Cela a un sens parce que normalement, l'URI de demande va identifier un UAS pour les demandes de mi dialogue. Dans ces cas, les décisions d'acheminement ont déjà été prises sur la demande initiale, et il n'y aurait aucun sens à les reprendre pour les demandes suivantes dans le dialogue.

5. Comportement de l'UAC

Un appelant qui souhaite exprimer des préférences pour une demande inclut des champs d'en-tête Accept-Contact, Reject-Contact, ou Request-Disposition dans la demande, selon ses préférences particulières. Aucun comportement supplémentaire n'est requis après l'envoi de la demande.

Les champs d'en-tête Accept-Contact, Reject-Contact, et Request-Disposition dans un ACK pour une réponse finale non 2xx, ou dans une demande CANCEL, DOIVENT être égaux aux valeurs dans la demande originale acquittée ou annulée. Cela est destiné à assurer un fonctionnement approprié à travers les mandataires sans état.

Si l'UAC veut déterminer si les serveurs le long du chemin comprennent les champs d'en-tête décrits dans la présente spécification, il inclut un champ d'en-tête Proxy-Require avec une valeur de "pref" [RFC3840] dans sa demande. Si la demande devait échouer avec un code de réponse de 420, l'UAC saurait que l'extension n'est pas acceptée. Dans ce cas, il DEVRAIT réessayer, et peut décider si il utilise ou non les préférences de l'appelant. Un UA ne devrait utiliser Proxy-Require que si la connaissance de la prise en charge est essentielle pour le traitement de la demande. Noter que dans tous les cas, les préférences de l'appelant ne sont considérées que comme des préférences – il n'est pas garanti que le service demandé sera exécuté. À ce titre, l'inclusion d'un champ d'en-tête Proxy-Require ne signifie pas que les préférences seront exécutées, mais juste que l'extension Préférences de l'appelant est comprise par les mandataires.

5.1 Préférences de traitement de la demande

Le champ d'en-tête Request-Disposition spécifie les préférences de l'appelant pour la façon dont un serveur devrait traiter une demande. Sa valeur est une liste de jetons, dont chacun spécifie une directive de traitement particulière.

La syntaxe du champ d'en-tête se trouve à la Section 10, et la sémantique des directives est décrite au paragraphe 9.1.

5.2 Préférences d'ensemble de caractéristiques

Un UAC peut indiquer les préférences de l'appelant pour les capacités d'un UA qui devrait être ou non atteint par suite de l'envoi d'une demande SIP. Pour ce faire, il ajoute une ou plusieurs valeurs de champ d'en-tête Accept-Contact et Reject-Contact. Chaque valeur de champ d'en-tête contient un ensemble de paramètres de caractéristiques qui définissent un ensemble de caractéristiques. La syntaxe du champ d'en-tête se trouve à la Section 10, et un exposé de son utilisation au paragraphe 9.2.

Chaque ensemble de caractéristiques est construit comme décrit à la Section 5 de la [RFC3840]. Les ensembles de caractéristiques placés dans ces champs d'en-tête PEUVENT se chevaucher ; c'est-à-dire qu'un UA PEUT indiquer des préférences pour des ensembles de caractéristiques qui se correspondent, conformément à l'algorithme de correspondance de la [RFC2533].

Un UAC peut exprimer des préférences explicites pour les méthodes et paquetages d'événements pris en charge par un UA. Il est RECOMMANDÉ qu'un UA comporte un terme dans un ensemble de caractéristiques Accept-Contact avec l'étiquette de caractéristique "sip.methods" (noter cependant que même si le nom de cette étiquette de caractéristique est sip.methods, il sera codé dans le champ d'en-tête Accept-Contact simplement comme "methods") dont la valeur comporte la méthode de la demande. Lorsque un UA envoie une demande SUBSCRIBE, il est RECOMMANDÉ qu'elle comporte un terme dans un ensemble de caractéristiques Accept-Contact avec l'étiquette de caractéristique "sip.events", dont la valeur inclut le paquetage d'événement de la demande. Il est à la discrétion de la mise en œuvre de décider si ces termes sont placés dans un nouvel ensemble de caractéristiques ou si ils sont inclus dans chaque ensemble de caractéristiques. Dans la plupart des cas, le bon effet est réalisé en incluant un terme dans chaque ensemble de caractéristiques.

Par exemple, le champ d'en-tête Accept-Contact suivant exprime le désir d'acheminer un appel à un appareil mobile, en utilisant des paramètres de caractéristique tirés de la [RFC3840] :

```
Accept-Contact: *,mobility="mobile";methods="INVITE"
```

Le champ d'en-tête Reject-Contact permet à l'UAC de spécifier qu'un UA ne devrait pas être contacté si il correspond à une des valeurs du champ d'en-tête. Chaque valeur du champ d'en-tête Reject-Contact contient une "*", simplement pour mettre la syntaxe en ligne avec les directives pour les extensions SIP de la [RFC4485], et est paramétrée par un ensemble de paramètres de caractéristiques. Tous les UA dont les capacités correspondent à l'ensemble de caractéristiques décrit par les paramètres de caractéristiques correspondent à la valeur.

Le champ d'en-tête Accept-Contact permet à l'UAC de spécifier qu'un UA devrait être contacté si il correspond à certaines ou à toutes les valeurs du champ d'en-tête. Chaque valeur du champ d'en-tête Accept-Contact contient une "*", et est paramétré par un ensemble de paramètres de caractéristiques. Tout UA dont les capacités correspondent à l'ensemble de caractéristiques décrit par les paramètres de caractéristiques correspond à la valeur. Le comportement précis dépend largement de la présence des paramètres "require" et "explicit". Lorsque ils sont tous les deux présents, un mandataire va seulement transmettre la demande aux contacts qui ont explicitement indiqué qu'ils prennent en charge l'ensemble de caractéristiques désiré. Tous les autres sont éliminés. À ce titre, un UAC ne devrait utiliser "require" et "explicit" ensemble que lorsque il souhaite que l'appel échoue, sauf si un contact correspond définitivement. Il est possible qu'un UA prenne en charge une caractéristique désirée, mais ne l'ait pas indiqué dans son enregistrement. Lorsque un UAC utilise à la fois "explicit" et "require", un tel contact ne sera pas atteint. Par suite, cette combinaison n'est souvent pas celle que va vouloir un UAC.

Lorsque seul "require" est présent, cela signifie qu'un contact ne sera pas utilisé si il ne correspond pas. Si il correspond, ou si on ne sait pas si c'est une correspondance complète, le contact est quand même utilisé. Un UAC utiliserait "require" seul lorsque un contact sans correspondance serait sans utilité. Cela est courant dans les services où la demande ne peut simplement pas être servie sans les caractéristiques nécessaires. Un exemple est la prise en charge de méthodes ou de paquetages d'événement spécifiques. Lorsque seul "require" est présent, le mandataire va aussi acheminer de préférence la demande sur l'UA qui représente la "meilleure" correspondance. Ici, "meilleure" signifie que l'UA a explicitement indiqué qu'il prend en charge plus des caractéristiques désirées qu'aucun autre. Noter cependant que cet acheminement préférentiel ne va jamais outrepasser un ordre fourni par la partie appelante. L'acheminement préférentiel va seulement choisir parmi les contacts de q-value égale.

Lorsque seul "explicit" est présent, cela signifie que tous les contacts fournis par l'appelé seront utilisés. Cependant, si le contact n'est pas une correspondance explicite, il est essayé en dernier parmi tous les autres contacts qui ont la même q-value. La principale différence entre cette configuration et l'utilisation de "require" et "explicit" est donc le comportement de repli pour les contacts qui ne correspondent pas explicitement. Ici, ils sont essayés en dernier recours. Si "require" est aussi présent, ils ne sont jamais essayés.

Finalement, si ni "require" ni "explicit" n'est présent, cela signifie que tous les contacts fournis par l'appelé seront utilisés. Cependant, si le contact ne correspond pas, il est essayé en dernier parmi tous les autres contacts qui ont la même q-value. Si il correspond, la demande est acheminée de préférence sur la "meilleure" correspondance. C'est une configuration courante pour les préférences qui, si elle n'est pas respectée, va quant même permettre la réussite de l'appel, et plus il y a de correspondance, mieux c'est.

6. Comportement de l'UAS

Lorsque un UAS conforme à la présente spécification reçoit une demande dont l'URI de demande correspond à un de ses contacts enregistrés, il DEVRAIT appliquer le comportement décrit au paragraphe 7.2 comme si il était un mandataire pour le domaine dans l'URI de demande. L'UAS agit comme si sa base de données de localisation contenait une seule cible de demande pour l'URI de demande. Cette cible est associée à un ensemble de caractéristiques. L'ensemble de caractéristiques est le même que celui placé dans l'enregistrement de l'URI dans l'URI de demande. Si un UA s'est enregistré sur plusieurs adresses d'enregistrement séparées, et si les contacts enregistrés pour chacune ont des capacités différentes, il aura utilisé un URI différent dans chaque enregistrement, de sorte qu'il peut déterminer quel ensemble de caractéristiques utiliser.

Ce traitement survient après que le client a authentifié et autorisé la demande, mais avant le reste du traitement général d'UAS décrit au paragraphe 8.2.1 de la [RFC3261].

Si, après avoir effectué ce traitement, il ne reste plus d'URI dans l'ensemble cible, l'UA DEVRAIT rejeter la demande avec une réponse 480. Si il reste un URI (il y en avait seulement un pour commencer) l'UA continue le traitement de la demande conformément à la [RFC3261].

Avoir un UAS qui effectue les opérations de correspondance comme si il était un mandataire permet d'honorer certaines des préférences de l'appelant, même si le mandataire ne prend pas en charge l'extension.

Un UAS DEVRAIT traiter toute directive de mise en file d'attente présente dans un champ d'en-tête Request-Disposition de la demande. Toutes les autres directives DOIVENT être ignorées.

7. Comportement du mandataire

Le comportement du mandataire consiste en deux jeux de règles orthogonaux – l'un pour le traitement du champ d'en-tête Request-Disposition, et l'autre pour le traitement de l'URI et des préférences de jeu de caractéristiques dans les champs d'en-tête Accept-Contact et Reject-Contact.

En plus du traitement de ces en-têtes, un mandataire PEUT en ajouter un s'il n'est pas présent, ou ajouter une valeur à un champ d'en-tête existant, comme si il était un UAC. Ceci est utile pour un mandataire pour demander le traitement chez les mandataires en aval dans la mise en œuvre d'une caractéristique. Cependant, un mandataire NE DOIT PAS modifier ou retirer une valeur de champ d'en-tête existante. Cela est particulièrement important lorsque S/MIME est utilisé. La signature du message pourrait inclure les champs d'en-tête de préférences de l'appelant, permettant à l'UAS de vérifier que, même si les mandataires ont ajouté des champs d'en-tête, les préférences de l'appelant originales sont toujours présentes.

7.1 Traitement de Request-Disposition

Si la demande contient un champ d'en-tête Request-Disposition et si c'est le propriétaire du domaine dans l'URI de demande, le serveur DEVRAIT exécuter les directives comme décrit au paragraphe 9.1, sauf si sa politique locale est configurée pour l'acheminer autrement.

7.2 Correspondance de préférence et de capacité

Un mandataire conforme à la présente spécification NE DOIT PAS appliquer à une demande l'opération de correspondance aux préférences décrite ici sauf si il est le propriétaire du domaine de l'URI de demande, et si il accède à un service de localisation qui a des capacités associées aux cibles de demande. Cependant, si il est le propriétaire du domaine, et si il

accède à un service de localisation qui a des capacités associées aux cibles de demande, il DEVRAIT appliquer le traitement décrit dans cette section. Normalement, c'est un mandataire qui utilise une base de données d'enregistrement pour déterminer les cibles de demande. Cependant, si un mandataire connaît les capacités par d'autres moyens, il DEVRAIT aussi appliquer le traitement défini ici. Si il effectue bien le traitement, il DOIT le faire comme décrit ci-dessous.

Le traitement est décrit par une conversion de la syntaxe décrite dans la présente spécification en la syntaxe de la [RFC2533], suivie par une opération de correspondance et un tri des valeurs de contact résultantes. L'usage de la syntaxe de la [RFC2533] comme étape intermédiaire n'est pas exigé ; c'est seulement un outil commode pour décrire le comportement exigé du mandataire. Un mandataire peut utiliser toutes les étapes qu'il veut du moment que le résultat est identique à celui qui serait obtenu par le traitement décrit ici.

7.2.1 Extraction des préférences explicites

La première étape du traitement de mandataire est d'extraire les préférences explicites. Pour ce faire, il recherche les champs d'en-tête Accept-Contact et Reject-Contact.

Pour chaque valeur de ces champs d'en-tête, il extrait les paramètres de caractéristiques. Ce sont les paramètres de champ d'en-tête dont les noms sont "audio", "automata", "class", "duplex", "data", "control", "mobility", "description", "events", "priority", "methods", "extensions", "schemes", "application", "video", "language", "type", "isfocus", "actor", ou "text", ou dont le nom commence par un plus (+) [RFC3840]. Le mandataire convertit tous ces paramètres dans la syntaxe de la [RFC2533], sur la base des règles de la Section 8.

Le résultat sera un ensemble de prédicats d'ensemble de caractéristiques en forme normale conjonctive, dont chacun est associé à un des deux champs d'en-tête de préférences. Si un req-parameter est associé à une valeur de champ d'en-tête dans le champ d'en-tête Accept-Contact, le prédicat d'ensemble de caractéristiques déduit de cette valeur de champ d'en-tête est dit avoir son fanion require établi. De même, si il y avait un explicit-param associé à une valeur de champ d'en-tête dans le champ d'en-tête Accept-Contact, le prédicat d'ensemble de caractéristiques déduit de cette valeur de champ d'en-tête est dit avoir son fanion explicit établi.

7.2.2 Extraction des préférences implicites

Si, et seulement si, le mandataire n'a pas trouvé de préférences explicites dans la demande (parce qu'il n'y avait pas de champ d'en-tête Accept-Contact ou Reject-Contact) le mandataire extrait les préférences implicites. Ces préférences sont celles qui sont impliquées par la présence d'autres informations dans la demande.

D'abord, le mandataire crée une conjonction sans termes. Cette conjonction représente un ensemble de caractéristiques qui vont être associées au champ d'en-tête Accept-Contact, comme si elles y étaient incluses. Noter que cela n'implique pas de modification du message – seulement une association pour les besoins du traitement. De plus, cet ensemble de caractéristiques a son fanion require établi, mais pas son fanion explicit.

Le mandataire ajoute alors des termes à la conjonction pour les deux types de préférences implicites ci-dessous.

7.2.2.1 Méthodes

Une préférence implicite est la méthode. Lorsque un UAC envoie une demande avec une méthode spécifique, il a une préférence implicite pour que la demande soit acheminée seulement aux UA qui acceptent cette méthode. Pour prendre en charge cette préférence implicite, le mandataire ajoute un terme à la conjonction de la forme suivante :

(sip.methods=[méthode de demande])

7.2.2.2 Paquetages d'événement

Pour les demandes qui établissent un abonnement [RFC3265], le champ d'en-tête Event est une autre expression d'une préférence implicite. Il exprime le désir que la demande ne soit acheminée qu'à un serveur qui prend en charge le paquetage d'événements en question. Pour prendre en charge cette préférence implicite, le mandataire ajoute un terme à la conjonction de la forme suivante :

(sip.events=[valeur du champ d'en-tête Event])

7.2.3 Construction des prédicats de contact

Le mandataire prend alors chaque URI de l'ensemble cible (l'ensemble d'URI sur lesquels il va mandater ou rediriger) et il obtient ses capacités comme un prédicat d'ensemble de caractéristiques dans le format de la [RFC2533]. C'est appelé un prédicat de contact. Si l'URI cible a été obtenu par un enregistrement, le mandataire calcule le prédicat de contact en extrayant les paramètres de caractéristiques du champ d'en-tête Contact [RFC3840] et en les convertissant ensuite en un prédicat de caractéristiques. Pour extraire les paramètres de caractéristiques, le mandataire suit les étapes suivantes :

1. Créer une liste initiale vide de paramètres de caractéristiques.
2. Si les paramètres de l'URI Contact comportent les paramètres "audio", "automata", "class", "duplex", "data", "control", "mobility", "description", "events", "priority", "methods", "schemes", "application", "video", "actor", "language", "isfocus", "type", "extensions", ou "text", ils sont copiés dans la liste.
3. Si un nom de paramètre d'URI Contact commence par un "+", il est copié dans la liste si la liste ne contient pas déjà ce nom en retirant le plus. En d'autres termes, si le paramètre de caractéristique "video" est dans la liste, le paramètre "+video" ne sera pas placé dans la liste. Ce conflit ne devrait jamais arriver si le client est conforme à la [RFC3840], car il est illégal d'utiliser la forme + pour coder une étiquette de caractéristique dans l'ensemble de base.

Si l'URI dans l'ensemble cible n'a pas de paramètre de caractéristiques, il est dit être immune au traitement des préférences de l'appelant. Cela signifie que l'URI est retiré temporairement de l'ensemble cible, que le traitement des préférences de l'appelant décrit ci-dessous est exécuté, et qu'ensuite l'URI est rajouté.

En supposant que l'URI a des paramètres de caractéristiques, ils sont convertis à la syntaxe de la [RFC2533] en utilisant les règles de la Section 8.

Le prédicat résultant est associé à une q-value. Si le prédicat de contact a été appris par une demande REGISTER, la q-value est égale à la q-value dans le paramètre du champ d'en-tête Contact, autrement, "1.0" si elle n'est pas spécifiée.

Par exemple, considérons le champ d'en-tête Contact enregistré suivant :

```
Contact: <sip:user@example.com>;audio;video;mobility="fixed";+sip.message="TRUE";other-param=66372;
methods="INVITE,OPTIONS,BYE,CANCEL,ACK";schemes="sip,http"
```

Cela serait converti en le prédicat suivant :

```
(& (sip.audio=TRUE)
(sip.video=TRUE)
(sip.mobility=fixed)
(sip.message=TRUE)
(| (sip.methods=INVITE) (sip.methods=OPTIONS) (sip.methods=BYE) (sip.methods=CANCEL) (sip.methods=ACK))
(| (sip.schemes=sip) (sip.schemes=http)))
```

Noter que "other-param" n'a pas été considéré comme un paramètre de caractéristique, car il n'est pas une étiquette de base ni ne commence par un + en tête.

7.2.4 Correspondance

Il est important de noter que le mandataire n'a pas à savoir quoi que ce soit sur la signification des étiquettes de caractéristique qu'il compare afin d'effectuer l'opération de correspondance. Les règles pour effectuer la comparaison dépendent des indications syntaxiques présentes dans les valeurs de chaque étiquette de caractéristique. Par exemple, un prédicat tel que :

```
(foo>=4)
```

implique que l'étiquette de caractéristique "foo" est une valeur numérique. Les règles de correspondance de la [RFC2533] exigent seulement d'une mise en œuvre qu'elle sache si l'étiquette de caractéristique est un jeton numérique, ou une chaîne guillemetée (les booléens peuvent être traités comme des jetons). Les chaînes guillemetées sont toujours comparées en utilisant une opération de comparaison sensible à la casse. Les jetons sont comparés en utilisant une correspondance insensible à la casse. Ces deux cas sont différenciés par la présence de crochets angulaires autour de la valeur de l'étiquette de caractéristique. Lorsque ces crochets sont présents (c'est-à-dire, ;+sip.foo="<valeur>") cela implique une comparaison de chaîne sensible à la casse. Lorsque ils ne sont pas présents, (c'est-à-dire, (;+sip.bar="val") cela implique l'insensibilité à la casse. La comparaison des valeurs numériques utilise les comparaisons mathématiques normales.

D'abord, le mandataire applique les prédicats associés au champ d'en-tête Reject-Contact.

Pour chaque prédicat de contact, chaque prédicat Reject-Contact (c'est-à-dire, chaque prédicat associé au champ d'en-tête Reject-Contact) est examiné. Si ce prédicat Reject-Contact contient un filtre pour une étiquette de caractéristique, et si cette étiquette de caractéristique n'est pas présente quelque part dans le prédicat de contact, ce prédicat Reject-Contact est éliminé pour le traitement de ce prédicat de contact. Si le prédicat Reject-Contact n'est pas éliminé, il est confronté au prédicat de contact en utilisant l'opération de correspondance de la [RFC2533]. si le résultat est une correspondance, l'URI qui correspond à ce prédicat de contact est éliminé de l'ensemble cible.

Le résultat est que Reject-Contact va seulement éliminer les URI où l'UA a explicitement indiqué la prise en charge des caractéristiques qui ne sont pas voulues.

Ensuite, le mandataire applique les prédicats associés au champ d'en-tête Accept-Contact. Pour chaque contact qui reste dans l'ensemble cible, le mandataire construit un ensemble de correspondance, Ms. Initialement, cet ensemble contient tous les prédicats Accept-Contact. Chacun de ces prédicats est examiné. Ils sont confrontés au prédicat de contact en utilisant l'opération de correspondance de la [RFC2533]. Si le résultat n'est pas une correspondance, et si le prédicat Accept-Contact avait son fanion require établi, l'URI qui correspond à ce prédicat de contact est éliminé de l'ensemble cible. Si le résultat n'est pas une correspondance, mais si le prédicat Accept-Contact n'avait pas son fanion require établi, cet URI de contact n'est pas éliminé de l'ensemble cible, cependant, le prédicat Accept-Contact est retiré de l'ensemble de correspondance pour ce contact.

Pour chaque contact qui reste dans l'ensemble cible, le mandataire calcule un score pour ce contact par rapport à chaque prédicat de l'ensemble de correspondance du contact. Soit N le nombre de termes dans la conjonction de prédicats Accept-Contact. Chaque terme dans ce prédicat contient une seule étiquette de caractéristique. Si le prédicat de contact a un terme qui contient cette même étiquette de caractéristique, le score est incrémenté de 1/N. Si l'étiquette de caractéristique n'était pas présente dans le prédicat de contact, le score reste inchangé. Sur la base de ces règles, le score peut varier de zéro à un.

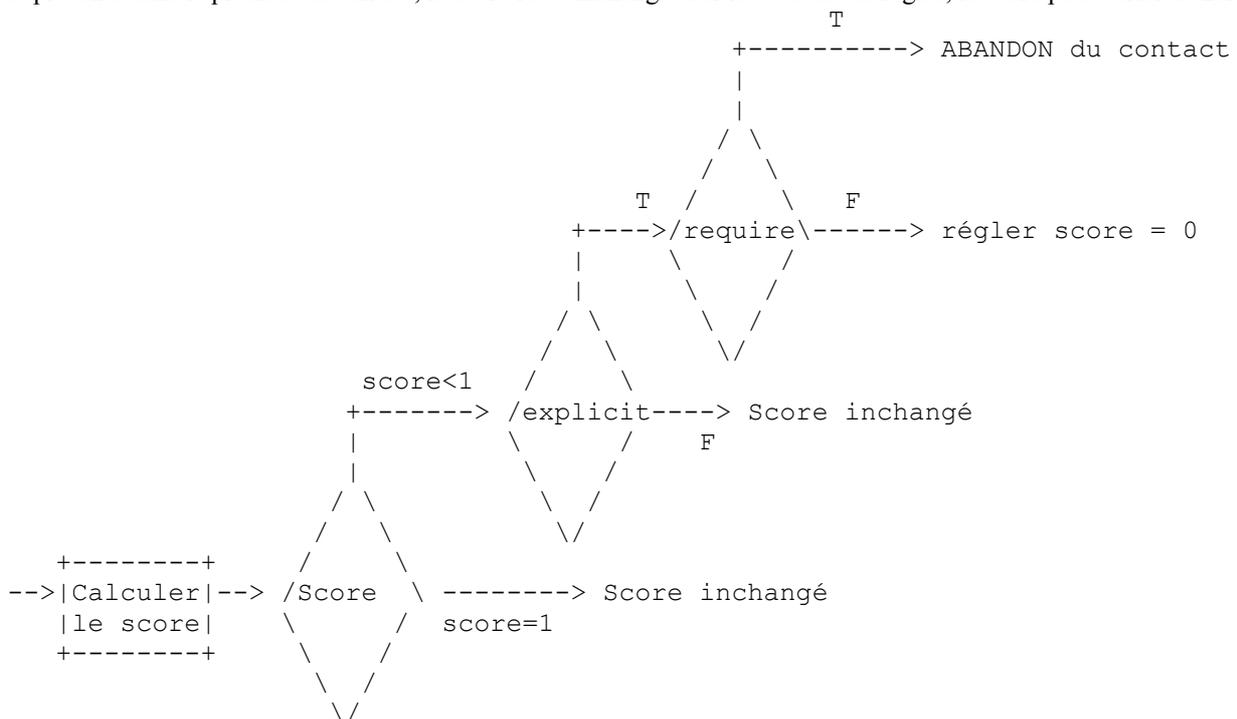


Figure 1 : Application du score

Les étiquettes require et explicit sont alors appliquées, résultant en modifications potentielles du score et de l'ensemble cible. Ce processus est résumé à la Figure 1. Si le score du prédicat de contact pour le prédicat Accept-Contact est inférieur à un, le prédicat Accept-Contact a une étiquette explicit, et si le prédicat a aussi une étiquette require, l'URI de contact qui correspond à ce prédicat de contact est abandonné. Cependant, si le prédicat n'avait pas d'étiquette require, le score est réglé à zéro. Si il n'y a pas d'étiquette explicit, le score est inchangé.

L'étape suivante est de combiner les scores et les q-values associées aux prédicats dans l'ensemble de correspondance, pour arriver à une préférence d'appelant globale, Q_a . Pour les URI qui restent dans l'ensemble cible, il y aura un score qui indique sa correspondance à l'égard de chaque prédicat Accept-Contact dans l'ensemble de correspondance. Si il y a M prédicats Accept-Contact dans l'ensemble de correspondance, il y aura M scores de S_1 à S_M , pour chaque contact. La préférence d'appelant globale, Q_a , est la moyenne arithmétique de S_1 à S_M .

À ce point est rajouté tout URI retiré de l'ensemble cible parce qu'immune aux préférences de l'appelant, et Qa pour cet URI est réglé à 1.0.

L'objet de la préférence d'appelant Qa est de fournir un ordre pour tous les contacts qui restent dans l'ensemble cible, si l'appelant n'a pas fourni un ordre. Pour ce faire, les contacts restants dans l'ensemble cible sont triés par q-value fournie par l'appelé. Une fois triés, ils sont groupés en classes d'équivalence, de sorte que tous les contacts avec la même q-value soient dans la même classe d'équivalence. Au sein de chaque classe d'équivalence, les contacts sont alors rangés selon leur valeur de Qa. Le résultat est une liste ordonnée des contacts qui est utilisée par le mandataire.

Si il n'y avait pas d'URI dans l'ensemble cible après l'application du traitement de ce paragraphe, et si les préférences de l'appelant étaient fondées sur des préférences implicites (paragraphe 7.2.2), le traitement de ce paragraphe est éliminé, et on utilise l'ensemble cible d'origine, ordonné selon les q-values d'origine.

Cela traite le cas où les préférences implicites pour la méthode ou paquetage d'événement résultaient en l'élimination de toutes les cibles potentielles. En revenant à l'ensemble cible d'origine, ces URI vont être essayés, et il va en résulter la génération d'une réponse 405 ou 489. L'UAC peut alors utiliser ces informations pour réessayer, ou faire rapport de l'erreur à l'utilisateur. Sans revenir à l'ensemble cible original, l'UAC verrait une réponse 480, et n'aurait pas connaissance de la raison de l'échec de la demande. Bien sûr, l'ensemble cible peut aussi être vide après l'application de préférences explicites. Il va en résulter la génération d'une réponse 480 par le mandataire. Ce comportement est acceptable, et en fait, désirable dans le cas de préférences explicites. Lorsque l'appelant fait une préférence explicite, il est d'accord pour que sa demande échoue si sa préférence ne peut être satisfaite. On peut essayer de retourner une erreur pour indiquer les capacités de l'appelé, afin que l'appelant puisse, peut-être, réessayer. Cependant, faire ainsi résulte en la fuite d'informations potentiellement sensibles vers l'appelant sans autorisation de l'appelé, et la présente spécification ne fournit donc aucun moyen pour cela.

Si un serveur mandataire est répétitif, il ajoute les champs d'en-tête Contact retournés dans les réponses redirect à l'ensemble cible, et applique à nouveau l'algorithme de préférences de l'appelant.

Si le serveur redirige, il retourne toutes les entrées dans l'ensemble cible. Il alloue les q-values à ces entrées afin que l'ordre soit identique à celui déterminé par le traitement ci-dessus. Cependant, il NE DOIT PAS inclure les paramètres de caractéristiques pour les entrées dans l'ensemble cible. Si il l'a fait, le serveur mandataire amont va appliquer encore une fois les mêmes préférences de l'appelant, résultant en une double application de ces préférences. Si le serveur de redirection souhaite bien inclure les paramètres de caractéristiques dans le champ d'en-tête Contact, il DOIT rediriger en utilisant l'ensemble cible original et les q-values d'origine, avant l'application des préférences de l'appelant.

7.2.5 Exemple

Considérons l'exemple suivant, qui est outré mais illustre les divers composants du procès de comparaison. Il y a cinq contacts enregistrés pour sip:user@example.com. Ce sont :

```
Contact: sip:u1@h.example.com;audio;video;methods="INVITE,BYE";q=0.2
Contact: sip:u2@h.example.com;audio="FALSE";methods="INVITE";actor="msg-taker";q=0.2
Contact: sip:u3@h.example.com;audio;actor="msg-taker";methods="INVITE";video;q=0.3
Contact: sip:u4@h.example.com;audio;methods="INVITE,OPTIONS";q=0.2
Contact: sip:u5@h.example.com;q=0.5
```

Un INVITE envoyé à sip:user@example.com contenait les champs d'en-tête de préférences de l'appelant suivants :

```
Reject-Contact: *;actor="msg-taker";video
Accept-Contact: *;audio;require
Accept-Contact: *;video;explicit
Accept-Contact: *;methods="BYE";class="business";q=1.0
```

Il n'y a pas de préférence implicite dans cet exemple, parce que des préférences explicites sont fournies.

Le mandataire retire d'abord u5 de l'ensemble cible, car il est immune au traitement des préférences de l'appelant.

Ensuite, le mandataire traite le champ d'en-tête Reject-Contact. C'est une comparaison pour tous les quatre contacts restants, mais seulement une correspondance explicite pour u3. C'est parce que u3 est le seul qui a explicitement indiqué la prise en charge de la vidéo, et explicitement indiqué qu'il est un preneur de messages. Ainsi, u3 est éliminé, et les autres restent.

Ensuite, chacun des trois contacts restant est comparé à chacun des trois prédicats Accept-Contact. u1 est une correspondance pour tous les trois, obtenant un score de 1,0 pour les deux premiers prédicats, et 0,5 pour le troisième

(l'étiquette de caractéristique `methods` était présente dans le prédicat de contact, mais l'étiquette de classe ne l'était pas). `u2` ne correspond pas au premier prédicat. Comme ce prédicat a une étiquette `require`, `u2` est éliminé. `u4` correspond au premier prédicat, obtenant un score de 1,0. `u4` correspond au second prédicat, mais comme la correspondance n'est pas explicite (le score est en fait 0,0) le score est réglé à zéro (il était déjà de zéro, de sorte que rien ne change). `u4` ne correspond pas au troisième prédicat.

À ce point, `u1` et `u4` restent. `u1` correspondait aux trois prédicats `Accept-Contact`, de sorte que son ensemble de correspondance les contient tous trois, avec les scores de 1, 1, et 0,5. `u4` correspond aux deux premiers prédicats, avec les scores de 1,0 et 0,0. `Qa` pour `u1` est 0,83 et `Qa` pour `u4` est 0,5. `u5` est rajouté avec un `Qa` de 1,0.

Ensuite, les contacts restants dans l'ensemble cible sont triés par `q-value`. `u5` a une valeur de 0,5, `u1` a une `q-value` de 0,2 ainsi que `u4`. Il y a deux classes d'équivalence. La première à une `q-value` de 0,5, et comporte juste `u5`. Comme il y a seulement un membre de la classe, le tri au sein de la classe n'a pas d'impact. La seconde classe d'équivalence a une `q-value` de 0,2. Au sein de cette classe, les deux contacts, `u1` et `u4`, sont ordonnés sur la base de leur valeur de `Qa`. `u1` a un `Qa` de 0,83, et `u4`, un `Qa` de 0,5. Donc, `u1` vient en premier, suivi par `u4`. L'ensemble ordonné global résultant de contacts dans la cible est `u5`, `u1`, puis `u4`.

8. Transposition des paramètres de caractéristique en prédicat

La transposition entre les paramètres de caractéristiques et un prédicat d'ensemble de caractéristiques, formatée conformément à la syntaxe de la [RFC2533], est triviale. C'est juste l'opposé du processus décrit à la Section 5 de la [RFC3840].

En partant d'un ensemble de paramètres de caractéristiques, la procédure est la suivante. Construire une conjonction. Chaque terme de la conjonction découle d'un paramètre de caractéristique. Si le paramètre de caractéristique n'a pas de valeur, il est équivalent, dans les termes du traitement qui suit, à ce qu'il ait une valeur de "VRAI".

Si la valeur du paramètre de caractéristique est une liste de valeurs d'étiquette, l'élément de la conjonction est une disjonction. Il y a un terme dans la disjonction pour chaque valeur d'étiquette dans la liste de valeurs d'étiquette.

Considérons maintenant la construction d'un filtre à partir d'une valeur d'étiquette. Si la valeur d'étiquette commence par un point d'exclamation (!), le filtre est de la forme :

(! <filtre à partir du reste>)

où "<filtre à partir du reste>" se réfère au filtre qui aurait été construit à partir de la valeur d'étiquette si le point d'exclamation n'avait pas été présent.

Si la valeur d'étiquette commence par un dièse (#), le filtre est une comparaison numérique. Le comparateur est `=`, `>=`, `<=`, ou une gamme fondée sur le prochain caractère dans la phrase. Si le prochain caractère est `=`, `>=`, ou `<=`, le filtre est de la forme :

(nom comparateur compare-valeur)

où `nom` est le nom du paramètre de caractéristique après décodage (voir ci-dessous) et le comparateur est `=`, `>=`, ou `<=` selon les caractères initiaux de la phrase. Si le reste du texte dans la valeur d'étiquette après le signe égal contient une virgule décimale (impliquant un nombre rationnel) la virgule décimale est glissée `N` fois vers la droite jusqu'à ce qu'il soit un entier, `I`. `Compare-valeur` ci-dessus est alors réglé à "`I / 10**N`", où `10**N` est le résultat du calcul du nombre 10 à la puissance `N`.

Si la valeur après le dièse est un nombre, le filtre est une gamme. Le format du filtre est :

(nom=<reste>)

où "`nom`" est l'étiquette de caractéristique après décodage (voir ci-dessous) et "<reste>" est le reste du texte de la valeur d'étiquette après le #, avec tout nombre décimal converti en forme rationnelle, et le deux-points remplacé par un double point (..).

Si la valeur d'étiquette ne commence pas par un dièse (c'est un jeton `nobang` ou booléen) le filtre est de la forme :

(nom=valeur d'étiquette)

où nom est l'étiquette de caractéristique après décodage (voir ci-dessous).

Si le paramètre de caractéristique contient une valeur de chaîne (sur la base du fait qu'elle commence par un crochet angulaire gauche (" $<$ ") et se termine par un crochet angulaire droit (" $>$ ") le filtre est de la forme :

```
(nom="qdtype")
```

Noter l'usage explicite de guillemets autour de qdtype, qui indiquent que la valeur est une chaîne. Dans la [RFC2533], les chaînes sont comparées en utilisant une règle sensible à la casse, et les jetons sont comparés en utilisant des règles insensibles à la casse.

Les étiquettes de caractéristiques, comme spécifié dans la [RFC2506], ne peuvent pas être directement représentées comme paramètres de champ d'en-tête dans les champs d'en-tête Contact, Accept-Contact, et Reject-Contact. Cela est dû à une incohérence de la grammaire, et au besoin de différencier les paramètres de caractéristique des paramètres utilisés par d'autres extensions. À ce titre, les valeurs d'étiquette de caractéristique sont codées à partir du format de la RFC2506 pour donner une enc-feature-tag, et sont ensuite décodées dans le format de la RFC2506. Le processus de décodage est simple. Si il y a un signe plus (+) en tête, il est retiré. Tout point d'exclamation (!) est converti en deux-points (:), et tout guillemet simple (') est converti en barre oblique (/). Si il n'y avait pas de signe plus en tête, et si le reste du nom codé était "audio", "automata", "class", "duplex", "data", "control", "mobility", "description", "events", "priority", "methods", "schemes", "application", "video", "actor", "isfocus", "extensions" ou "text", le préfixe "sip." est ajouté au reste du nom codé pour calculer le nom d'étiquette de caractéristique.

Par exemple, le champ d'en-tête Accept-Contact :

```
Accept-Contact:*;mobility="fixed";events="!presence,message-summary";language="en,de";description="<PC>";
+sip.newparam;+rangeparam="#-4:+5.125"
```

serait converti en le prédicat de caractéristique suivant :

```
(& (sip.mobility=fixed)
(| (! (sip.events=presence)) (sip.events=message-summary))
(| (language=en) (language=de))
(sip.description="PC")
(sip.newparam=TRUE)
(rangeparam=-4..5125/1000))
```

9. Définitions de champ d'en-tête

La présente spécification définit trois nouveaux champs d'en-tête : Accept-Contact, Reject-Contact, et Request-Disposition.

La Figure 2 et la Figure 3 sont une extension des Tableaux 2 et 3 de la [RFC3261] pour les champs d'en-tête Accept-Contact, Reject-Contact, et Request-Disposition. La colonne "INF" est pour la méthode INFO [RFC2976], "PRA" est pour la méthode PRACK [RFC3262], "UPD" est pour la méthode UPDATE [RFC3311], "SUB" est pour la méthode SUBSCRIBE [RFC3265], "NOT" est pour la méthode NOTIFY [RFC3265], "MSG" est pour la méthode MESSAGE [RFC3428], et "REF" est pour la méthode REFER [RFC3515].

Champ d'en-tête	où	mandataire	ACK	BYE	CAN	INV	OPT	REG
Accept-Contact	R	ar	0	0	0	0	0	-
Reject-Contact	R	ar	0	0	0	0	0	-
Request-Disposition	R	a	0	0	0	0	0	0

Figure 2 : Champs d'en-tête Accept-Contact, Reject-Contact, et Request-Disposition

Champ d'en-tête	où	mandataire	PRA	UPD	SUB	NOT	INF	MSG	REF
Accept-Contact	R	a	0	0	0	0	0	0	0
Reject-Contact	R	ar	0	0	0	0	0	0	0
Request-Disposition	R	ar	0	0	0	0	0	0	0

Figure 3 : Champs d'en-tête Accept-Contact, Reject-Contact, et Request-Disposition

9.1. Request-Disposition

Le champ d'en-tête Request-Disposition spécifie les préférences de l'appelant sur la façon dont un serveur devrait traiter une demande. Sa valeur est une liste de jetons, dont chacun spécifie une certaine directive. Sa syntaxe est spécifiée à la Section 10. Noter qu'une forme compacte, qui utilise la lettre d, a été définie. Les directives sont groupées par type. Il peut seulement y avoir une directive de chaque type par demande (par exemple, on ne peut pas avoir à la fois "proxy" et "redirect" dans le même champ d'en-tête Request-Disposition).

Lorsque l'appelant spécifie une directive, le serveur DEVRAIT respecter cette directive.

Les types de directives suivants sont définis :

proxy-directive : Ce type de directive indique si l'appelant aimerait que chaque serveur mandate ("proxy") ou redirige ("redirect").

cancel-directive : Ce type de directive indique si l'appelant aimerait que chaque serveur mandataire envoie une demande CANCEL vers l'aval ("cancel") en réponse à un 200 OK provenant du serveur aval (qui est le mode de fonctionnement normal, le rendant redondant) ou si cette fonction devrait être laissée à l'appelant ("no-cancel"). Si un mandataire reçoit une demande avec ce paramètre réglé à "no-cancel", il NE DEVRAIT PAS annuler de branche en cours à réception d'une 2xx. Cependant, il va quand même envoyer CANCEL sur toute branche en cours à réception d'une 6xx.

fork-directive : Ce type de directive indique si un mandataire devrait diviser une demande ("fork"), ou mandater à une seule adresse ("no-fork"). Si il est demandé au serveur de ne pas diviser, le serveur DEVRAIT mandater la demande à la "meilleure" adresse (généralement celle de la plus forte q-value). Si il y a plusieurs adresses avec la plus forte q-value, le serveur choisit sur la base de sa politique locale. La directive est ignorée si "redirect" a été demandé.

recurse-directive : Ce type de directive indique si un serveur mandataire qui reçoit une réponse 3xx devrait envoyer des demandes aux adresses énumérées dans la réponse ("recurse") ou transmettre la liste des adresses en amont vers l'appelant ("no-recurse"). La directive est ignorée si "redirect" a été demandé.

parallel-directive : Pour un serveur mandataire qui divise, ce type de directive indique si l'appelant aimerait que le serveur mandataire mandate tout de suite la demande à toutes les adresses connues ("parallel") ou les passe à la suite les unes des autres, ne contactant l'adresse suivante qu'après avoir reçu une réponse finale non 2xx ou non 6xx pour la précédente ("sequential"). La directive est ignorée si "redirect" a été demandé.

queue-directive : Si l'appelant est temporairement injoignable, par exemple, parce que il est déjà occupé, l'appelant peut indiquer qu'il veut que son appel soit mis en file d'attente ("queue") ou rejeté immédiatement ("no-queue"). Si l'appel est mis en file d'attente, le serveur retourne "182 Queued". Un appel mis en file d'attente peut être terminé comme décrit dans la [RFC3261].

Exemple :

Request-Disposition: proxy, recurse, parallel

L'ensemble des directives request-disposition n'est pas extensible à volonté. Cela est fait pour éviter une prolifération de nouvelles extensions à SIP qui seraient "tunnelées" à travers ce champ d'en-tête.

9.2 Champs d'en-tête Accept-Contact et Reject-Contact

La syntaxe de ces champs d'en-tête est décrite à la Section 10. Une forme compacte, avec la lettre a, a été définie pour le champ d'en-tête Accept-Contact, et avec la lettre j pour le champ d'en-tête Reject-Contact.

10. BNF augmenté

Le BNF pour le champ d'en-tête Request-Disposition est :

```
Request-Disposition = ( "Request-Disposition" / "d" ) HCOLON directive *(COMMA directive)
directive           = proxy-directive / cancel-directive / fork-directive / recurse-directive / parallel-directive / queue-directive
proxy-directive     = "proxy" / "redirect"
cancel-directive    = "cancel" / "no-cancel"
fork-directive      = "fork" / "no-fork"
recurse-directive   = "recurse" / "no-recurse"
parallel-directive  = "parallel" / "sequential"
```

queue-directive = "queue" / "no-queue"

Le BNF pour les champs d'en-tête Accept-Contact et Reject-Contact est :

Accept-Contact = ("Accept-Contact" / "a") HCOLON ac-value *(COMMA ac-value)

Reject-Contact = ("Reject-Contact" / "j") HCOLON rc-value *(COMMA rc-value)

ac-value = "*" *(SEMI ac-params)

rc-value = "*" *(SEMI rc-params)

ac-params = feature-param / req-param / explicit-param / generic-param ; paramètre de caractéristique de la RFC3840

; paramètre générique de la RFC3261

rc-params = feature-param / generic-param

req-param = "require"

explicit-param = "explicit"

Malgré le BNF, il NE DOIT PAS y avoir plus d'un req-param ou explicit-param dans un ac-params. De plus, il ne peut y avoir qu'une seule instance de toute étiquette de caractéristique dans un paramètre de caractéristique (*feature-param*).

11. Considérations pour la sécurité

La présence des préférences de l'appelant dans une demande a un effet sur la façon dont la demande est traitée au serveur. Il en résulte que les demandes avec des préférences de l'appelant DEVRAIENT être protégées en intégrité avec les mécanismes sips spécifiés à la section 26 de la [RFC3261].

Le traitement des préférences de l'appelant exige des opérations d'établissement et les recherches peuvent exiger une certaine quantité de calculs. Cela permet une attaque de DOS par laquelle un usager peut envoyer des demandes avec un nombre substantiel de préférences de l'appelant, dans l'espoir de surcharger le serveur. Pour contrer cela, les serveurs DEVRAIENT rejeter les demandes qui ont trop de règles. Un nombre raisonnable est autour de 20.

12. Considérations relatives à l'IANA

La présente spécification enregistre trois nouveaux champs d'en-tête SIP, conformément au processus de la [RFC3261].

Voici l'enregistrement pour le champ d'en-tête Accept-Contact :

Numéro de RFC : RFC 3841

Nom du champ d'en-tête : Accept-Contact

Forme compacte : a

Voici l'enregistrement pour le champ d'en-tête Reject-Contact :

Numéro de RFC : RFC 3841

Nom de champ d'en-tête : Reject-Contact

Forme compacte : j

Voici l'enregistrement pour le champ d'en-tête Request-Disposition :

Numéro de RFC : RFC 3841

Nom de champ d'en-tête : Request-Disposition

Forme compacte : d

13. Remerciements

L'ensemble initial d'étiquettes de caractéristiques de support utilisé par la présente spécification a été influencé par la conception de CMA par Scott Petrack. Jonathan Lennox, Bob Penfield, Ben Campbell, Mary Barnes, Rohan Mahy, et John Hearty ont fourni des commentaires utiles. Graham Klyne a apporté son aide sur l'utilisation de la RFC 2533.

14. Références

14.1 Références normatives

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

- [RFC2533] G. Klyne, "Syntaxe de description des [ensembles de caractéristiques des supports](#)", mars 1999. (MàJ par [RFC2738](#), [RFC2938](#)) (P.S.)
- [RFC2976] S. Donovan, "Méthode INFO pour SIP", octobre 2000. (P.S., Remplacée par la RFC6086)
- [RFC3261] J. Rosenberg et autres, "SIP : [Protocole d'initialisation de session](#)", juin 2002. (Mise à jour par [RFC3265](#), [RFC3853](#), [RFC4320](#), [RFC4916](#), [RFC5393](#), RFC6665)
- [RFC3262] J. Rosenberg et H. Schulzrinne, "[Fiabilité des réponses provisoires](#) dans le protocole d'initialisation de session (SIP)", juin 2002. (P.S.)
- [RFC3265] A.B. Roach, "[Notification d'événement spécifique](#) du protocole d'initialisation de session (SIP)", juin 2002. (MàJ par [RFC6446](#)) (Remplacée par la RFC6665)
- [RFC3311] J. Rosenberg, "[Méthode UPDATE](#) du protocole d'initialisation de session (SIP) ", octobre 2002.
- [RFC3428] B. Campbell et autres, "[Extension de messagerie instantanée](#) pour le protocole d'initialisation de session (SIP) ", décembre 2002.
- [RFC3515] R. Sparks, "[Méthode Refer](#) du protocole d'initialisation de session (SIP)", avril 2003.
- [RFC3840] J. Rosenberg, H. Schulzrinne et P. Kyzivat, "[Indication des capacités d'agent d'utilisateur](#) dans le protocole d'initialisation de session (SIP)", août 2004

14.2 Références pour information

- [RFC2506] K. Holtman, A. Mutz, T. Hardie, "Procédure d'[enregistrement d'étiquette de caractéristique](#) de support", mars 1999. ([BCP0031](#))
- [RFC2824] J. Lennox et H. Schulzrinne, "Cadres et exigences du [langage de traitement d'appel](#) (CPL)", mai 2000.
- [RFC4485] J. Rosenberg, H. Schulzrinne, "Lignes directrices pour les auteurs d'extensions au protocole d'initialisation de session (SIP)", mai 2006. (*Information*)

15. Adresse des auteurs

Jonathan Rosenberg
dynamicsoft
600 Lanidex Plaza
Parsippany, NJ 07054
USA
téléphone : +1 973 952-5000
mél : jdrosen@dynamicsoft.com
URI : <http://www.jdrosen.net>

Henning Schulzrinne
Columbia University
M/S 0401
1214 Amsterdam Ave.
New York, NY 10027
USA
mél : schulzrinne@cs.columbia.edu
URI : <http://www.cs.columbia.edu/~hgs>

Paul Kyzivat
Cisco Systems
1414 Massachusetts Avenue
BXB500 C2-2
Boxboro, MA 01719
USA
mél : pkyzivat@cisco.com

16. Déclaration complète de droits de reproduction

Copyright (C) The Internet Society (2004). Le présent document est soumis aux droits, licences et restrictions contenus dans le BCP 78, et à www.rfc-editor.org, et sauf pour ce qui est mentionné ci-après, les auteurs conservent tous leurs droits.

Le présent document et les informations qui y sont contenues sont fournis 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.

Propriété intellectuelle

L'IETF ne prend pas position sur la validité et la portée de tout droit de propriété intellectuelle ou autres droits qui pourraient être revendiqués au titre de la mise en œuvre ou l'utilisation de la technologie décrite dans le présent document ou sur la mesure dans laquelle toute licence sur de tels droits pourrait être ou n'être pas disponible ; pas plus qu'elle ne prétend avoir accompli aucun effort pour identifier de tels droits. Les informations sur les procédures de l'ISOC au sujet des droits dans les documents de l'ISOC figurent dans les BCP 78 et BCP 79.

Des copies des dépôts d'IPR faites au secrétariat de l'IETF et toutes assurances de disponibilité de licences, ou le résultat de tentatives faites pour obtenir une licence ou permission générale d'utilisation de tels droits de propriété par ceux qui mettent en œuvre ou utilisent la présente spécification peuvent être obtenues sur le répertoire en ligne des IPR de l'IETF à <http://www.ietf.org/ipr> .

L'IETF invite toute partie intéressée à porter son attention sur tous copyrights, licences ou applications de licence, ou autres droits de propriété qui pourraient couvrir les technologies qui peuvent être nécessaires pour mettre en œuvre la présente norme. Prière d'adresser les informations à l'IETF à ietf- ipr@ietf.org .

Remerciement

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